Raccorder son réseau d'entreprise à l'Internet

Alexandre Fenyö, Frédéric Le Guern, Samuel Tardieu

Raccorder son réseau d'entreprise à l'Internet

Alexandre Fenyo
Frédéric Le Guern
Samuel Tardieu

Alexandre Fenyö - Frédéric Le Guern - Samuel Tardieu

Raccorder son réseau d'entreprise à l'Internet

- TCP/IP, routage IP
- Liaisons modem, RNIS, LS, X.25
- Serveurs DNS, SMTP, NNTP
- Serveurs Web : HTTP, HTML, CGI, proxy…
- Services IP Multicast : vidéo-conférence, travail coopératif
- Sécurité et firewalls

En cadeau !
Un CD-Rom contenant :

- **Logiciels et serveurs Unix : Apache, Spinner, BIND, sendmail, Satan…**
- **Logiciels clients : Eudora Light** *(PC/Mac)*, **Netscape Navigator 3.0** *(versions d'évaluation PC/Mac/Unix)…*

Eyrolles

Remerciements

Nous tenons tout particulièrement à remercier Philippe DAX, Nadine RICHARD et Ahmed SERHROUCHNI pour les précieux conseils qu'ils nous ont prodigués durant la rédaction de cet ouvrage ainsi que pour leur patience lors des multiples relectures.

Les noms d'utilisateurs fictifs présents dans les exemples de ce livre nous ont été gracieusement prêtés par Luc BEURTON et Stoned ELIPOT, qui nous ont également aidés par leurs pertinentes contributions.

Le chapitre sur le travail coopératif a bénéficié de l'aide de Yan PUJANTE, auteur d'un mémoire de recherche sur le sujet.

Nous tenons également à remercier Bruno BEAUGRAND, Raphaël LUTA et Philippe MEUNIER qui ont eu la gentillesse de nous procurer les documentations techniques dont nous avions besoin.

Pierre BEYSSAC et Guillaume URVOY nous ont également apporté leur aide et nous les en remercions.

Les illustrations présentant des matériels CISCO ont été reproduites avec la gracieuse autorisation de la société CISCO[1].

Le « formulaire de demande de création ou de modification de domaine dans la zone `fr` » a été reproduit avec l'aimable permission du NIC France[2], ainsi que la liste des fournisseurs proposant le service de raccordement par ligne spécialisée.

Nous remercions enfin la société INTUISYS[3] pour ses moyens techniques et la patience de ses collaborateurs.

1. `http://www.cisco.com/`
2. `http://www.nic.fr/`
3. `http://www.intuisys.fr/`

Préface

L'Internet est, de par sa topologie, un méta-réseau à couverture mondiale constitué d'agrégations de réseaux propriétaires qu'il interconnecte au moyen de lignes de transmission, de routeurs et du protocole d'interconnexion IP, l'Internet Protocol.

L'idée de cette interconnexion est déjà ancienne, puisqu'elle remonte à la naissance de l'Arpanet en 1969, lorsque Vinton Cerf raccorda les quatre premiers sites. Mais depuis, l'Internet n'a cessé de se développer et de s'amplifier à un rythme soutenu, doublant, sans fléchir, son taux de croissance chaque année.

Si ce déploiement n'a été expérimental qu'au cours des années 70 pour tester la viabilité des nouveaux protocoles, il est devenu plus concerté durant les années 80, uniquement réservé aux secteurs militaires et académiques ainsi qu'aux organismes et instituts de recherche et de développement où les raccordements point à point s'établissaient alors d'ami à ami. Au début des années 90 la déréglementation quasi générale des opérateurs de télécommunications allait ouvrir la voie à de nouvelles structures professionnelles offrant de la connectivité IP et des services réseau à valeur ajoutée à quiconque voulait adhérer au réseau.

Ces nouvelles structures, appelées ISP (Internet Service Provider) ou prestataire de service Internet, se sont ainsi multipliées pour faire face à la demande exponentielle de nouvelles populations issues des entreprises, mais aussi, par le biais de la médiatisation en quête de sensationnel, du grand public. Les ISP constituent dorénavant une couche incontournable entre les utilisateurs, toutes origines confondues, et l'Internet.

La première partie de cet ouvrage, consacrée au routage IP, c'est-à-dire au fonctionnement même de l'Internet, s'inscrit parfaitement dans cette optique d'interconnexion et de raccordement avec les prestataires de services. Elle présente tous les cas de figure de raccordements connus à ce jour et propose des solutions rationnelles selon les contraintes techniques et économiques. Les seconde et troisième parties du livre s'attachent principalement à développer les services majeurs du réseau qui sont actuellement présents sur l'Internet à savoir le courrier électronique *email*, les forums de discussion *news*, l'inévitable *World Wide Web*, et aussi les nouvelles technologies en cours d'expérimentation, telles que les relais, les caches et la diffusion de groupe par multicast. L'intérêt de ces chapitres porte sur l'installation, la mise en œuvre et la meilleure façon de rendre ces services. Les aspects de la sécurité sont abordés,

comme il se doit, dans la dernière partie du livre. L'administrateur système ou le responsable sécurité y trouvera un arsenal d'outils d'une grande efficacité destinés à protéger et à contrôler son site. Une grande majorité des services réseau et des outils de sécurité présentés est d'ailleurs, chose cocasse, apportée par l'utilisation de logiciels du domaine public très largement répandus et parfaitement maîtrisés par la communauté des internautes.

Ce livre s'attache à donner au responsable informatique et réseau de l'entreprise tous les éléments techniques pour à la fois comprendre, choisir et apporter des solutions concrètes et fiables à tous les problèmes que peuvent poser un raccordement à l'Internet et la mise en œuvre des services. Le lecteur pourra ainsi éviter d'une part de longues pertes de temps dans les méandres des documentations, d'autre part les écueils dans lesquels les premiers praticiens sont inévitablement tombés un jour. Cet ouvrage d'ingénierie s'adresse principalement aux administrateurs système et réseau, aux ingénieurs et responsables informatiques et bien entendu aux étudiants qui souhaitent toujours en savoir plus.

Ce travail original consacre le résultat de la collaboration d'un groupe de trois de mes élèves de l'ENST, promotions 1993 et 1994, qui ont chacun l'âge de l'Internet. J'ai personnellement eu avec chacun d'eux de longues discussions passionnantes à travers lesquelles l'échange à toujours primé et a été fructueux pour tous, sorte de maiëtique. La relève de la génération des pionniers par cette nouvelle « génération Internet » est donc assurée et prometteuse pour l'avenir à l'aube de l'an 2000.

Philippe DAX

Philippe DAX *est directeur d'études et ingénieur système et réseau du département informatique de l'École nationale supérieure des télécommunications de Paris (ENST), membre fondateur de l'Association française des utilisateurs d'Unix (AFUU), membre de l'ISOC (Internet SOCiety) France, auteur de nombreux articles à propos d'Internet, rédacteur en chef de la revue* Tribunix *et auteur du best-seller* Le langage C *traduit en de nombreuses langues (Eyrolles, 1983).*

Table des matières

PREMIÈRE PARTIE

Raccordement à l'Internet

Avant de pouvoir utiliser ou fournir des services sur l'Internet, il faut choisir une technique afin d'y raccorder son réseau d'entreprise : accès RTC par modem, RNIS avec routeur ou adaptateur de terminal, X25 à travers Transpac ou ligne spécialisée.

1

Introduction

Même si l'Internet est sous le feu des projecteurs dans les média depuis presque deux ans, il existait, sous une forme assez proche de ce qu'on connaît maintenant, déjà au début des années 70.

1.1 Historique et architecture de l'Internet

1.1.1 L'Internet : des années 60 à nos jours

À la fin des années 60, différents projets de réseaux à commutation de paquets ont vu le jour, dont Arpanet, le réseau de la Darpa (Defense Advanced Research Project Agency). C'est ce dernier qui a donné naissance à l'Internet. Pendant les années 70, le réseau s'est agrandi pour accueillir des centres du département de défense américain, le DoD, des sites universitaires et des centres de recherche. Au début des années 80, la famille des protocoles TCP/IP fut finalisée et la topologie de l'Internet qu'on connaît maintenant, ensemble de nœuds reliés par des routeurs IP, fut mise en place.

En 1985, Arpanet étant devenu complètement engorgé, la NSF (National Science Foundation) débuta la première phase du projet NSFNet. Ce dernier consistait à relier six centres informatiques majeurs et des réseaux universitaires par des liaisons à 56 Kbits/s.

En 1987, IBM, MCI et Merit[1] se virent attribuer pour cinq ans la tâche de mise à niveau et de gestion de la dorsale du réseau NSFNet, dorsale composée de liens à 1,5 Mbits/s faisant le tour des États-Unis et interconnectant principalement d'importants centres de calculs.

1. Merit est une organisation à but non lucratif regroupant onze universités de l'état du Michigan.

En 1990, ils se regroupèrent au sein d'ANS (Advanced Network and Services) pour administrer la dorsale du réseau NSFNet et faciliter la commercialisation de son accès. Merit fournit les moyens pour créer et administrer la Policy Routing Database, ou PRDB, base de données regroupant des informations de routage afin de configurer les routeurs majeurs du réseau pour acheminer correctement le trafic. ANS augmenta la capacité des liaisons jusqu'à 45 Mbits/s.

Toujours au début des années 1990, Sprint se chargea d'interconnecter la dorsale NSF avec de nombreux réseaux en Europe et en Asie.

L'administration de la dorsale NSF par Merit, établie en 1987 pour cinq ans et devant donc se terminer en 1992, fut reconduite pour fournir une période de transition afin de mettre en place la nouvelle architecture du réseau NSFNet :

- une dorsale à haut débit, le vBNS, ou *very high speed Backbone Network Service*, réseau de liaisons d'une capacité initiale de 155 Mbits/s, destiné uniquement au trafic entre les centres de recherche et les universités ;
- un ensemble de NAP, ou Network Access Points, situés en différents nœuds du vBNS, formant un ensemble de points de concentration et d'échange de trafic entre les fournisseurs Internet commerciaux ;
- un arbitre de routage, le Routing Arbiter, groupe chargé de mettre en place et d'administrer des bases de données prenant en compte les politiques de routage des fournisseurs Internet de la planète afin de configurer convenablement les échanges au niveau des NAP.

Début 1995, la transition du réseau NSFNet vers sa nouvelle architecture fut terminée et son administration par Merit prit fin. L'Internet venait donc d'entrer définitivement dans l'ère commerciale.

1.1.2 Topologie de l'Internet

Comme on peut le constater sur la figure 1.1 page ci-contre, l'Internet est maintenant constitué d'une dorsale vBNS qui fait le tour des États-Unis, interconnectant des réseaux régionaux ainsi que d'importants centres de calcul. Elle est reliée en cinq nœuds principaux à des points d'échange de trafic commercial sous forme de NAP ou d'autres types d'interconnexion, situés dans les régions de Chicago, Los Angeles, New York, San Francisco et Washington. Les fournisseurs Internet du monde entier viennent se connecter chacun sur plusieurs de ces points.

En Europe, on trouve différents réseaux fédérateurs comme par exemple une dorsale nommée Ebone, destinée à favoriser les échanges directs entre fournisseurs sans passer par les NAP américains, ou le réseau EuropaNET de DANTE (Delivery of Advanced Network Technology to Europe). EuropaNET est destiné au monde de la recherche tandis qu'Ebone propose aussi des accès aux fournisseurs commerciaux.

Plus de 60 fournisseurs Internet dans 29 pays sont interconnectés à travers Ebone, qui est

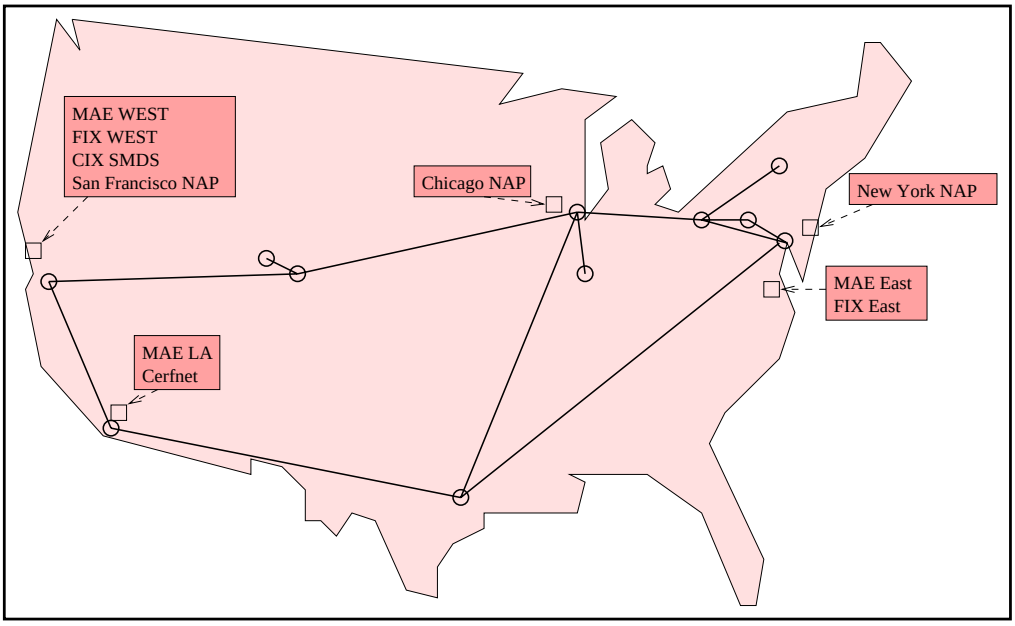

Figure 1.1 *Le vBNS et les NAP*

ainsi organisé autour de six villes reliées par des liaisons à quelques Mbits/s : Amsterdam, Stockholm, Munich, Paris, Vienne et Genève. Stockholm et Paris sont de plus reliées au réseau américain. Ainsi, les différents fournisseurs européens peuvent se connecter à un de ces points d'entrée d'Ebone pour acquérir une connectivité totale avec le reste des fournisseurs Internet du globe.

D'autres réseaux européens, tels que ceux de Pipex ou d'EUnet, sont interconnectés à Ebone. Ces réseaux privés n'utilisent pas Ebone pour accéder aux fournisseurs américains, ils disposent de leurs propres lignes transatlantiques, ce qui leur apporte l'indépendance et un meilleur contrôle de la qualité et de l'engorgement de leurs accès.

L'Internet Society (ISOC) met régulièrement à jour une carte de la connectivité Internet mondiale. Elle est présentée sur la figure 1.2 page suivante où on peut constater que la presque totalité de la planète est joignable à travers l'Internet.

1.1.3 Aspects administratifs

De nombreux organismes sont chargés d'une partie du travail administratif nécessaire au bon fonctionnement de l'Internet. On peut distinguer trois fonctions administratives de base :

- la gestion de l'arbitre de routage ;
- l'attribution des numéros (adresses IP, numéros de ports, etc.) ;
- l'attribution des noms de domaines.

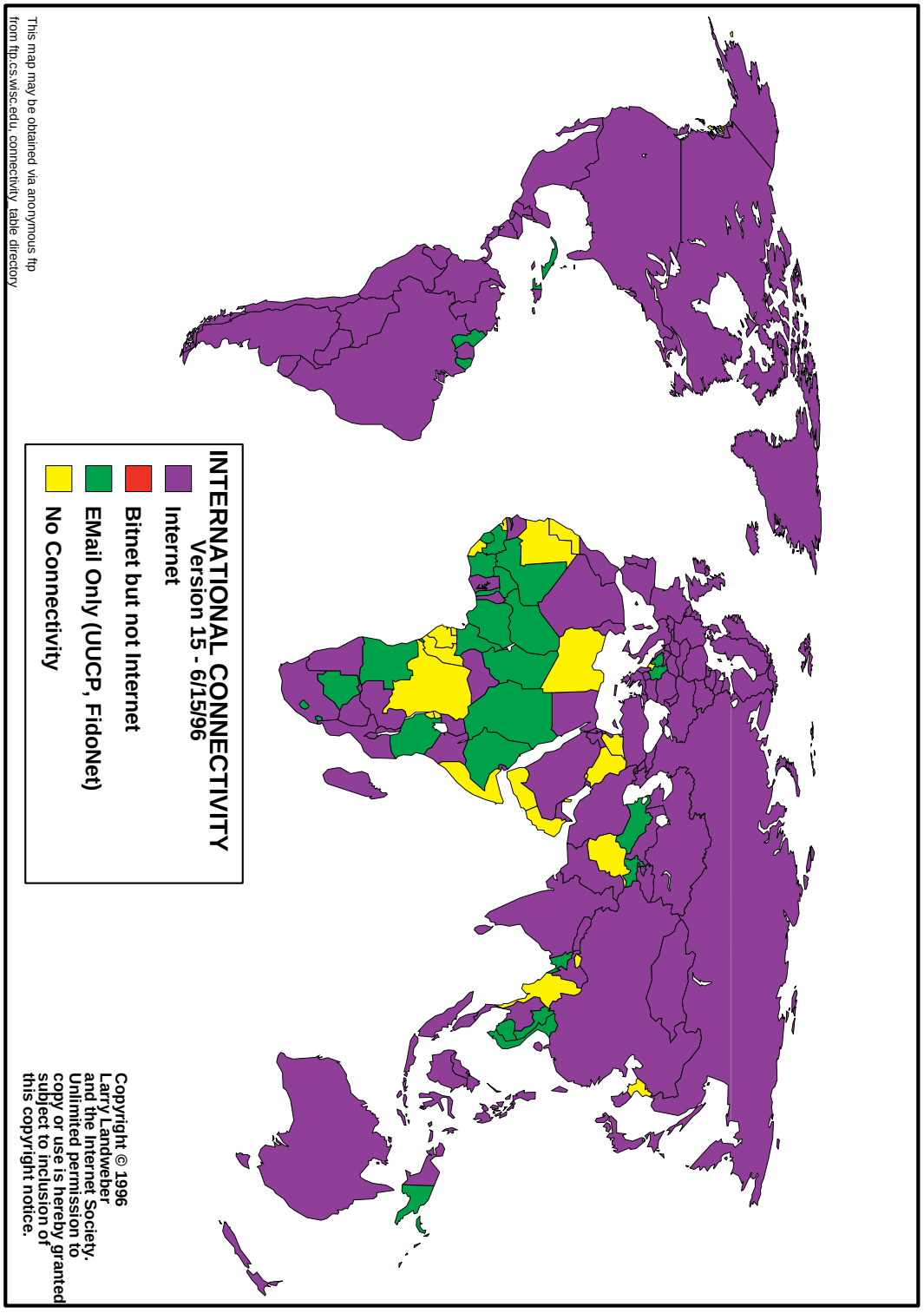

Figure 1.2 *Carte de la connectivité mondiale de l'Internet*

Gestion de l'arbitre de routage

Le projet d'arbitre de routage, ou Routing Arbiter Project, est pris en charge par Merit et l'Information Sciences Institute (ISI).

Les objectifs sont de normaliser un langage formel de description de politique de routage, de créer et distribuer des outils permettant de facilement évaluer et tester des politiques de routage, et de configurer les serveurs de routes (Route Servers) présents sur chacun des NAP en fonction des politiques des différents fournisseurs Internet.

Pour cela, des registres de routage Internet, ou Internet Routing Registries, sont chargés de collecter les différentes informations d'accessibilité de réseaux à partir des déclarations faites par les fournisseurs.

Il y a six registres de routage Internet :

- RIPE (Réseaux IP Européens) se charge de collecter les informations pour l'Europe ;
- MCI collecte les informations de ses clients ;
- CA*net collecte les informations de ses clients ;
- ANS collecte les informations de ses clients ;
- JPRR se charge de collecter les informations pour le Japon ;
- Merit et ISI collectent, au sein de la RADB (Routing Arbiter DataBase), les informations qui ne rentrent dans le cadre d'aucun des registres précédents.

Attribution des numéros

La gestion de tous les paramètres liés au fonctionnement de l'Internet a été déléguée par l'Internet Society à Iana : l'Internet Assigned Numbers Authority. Iana est actuellement pris en charge par l'Information Sciences Institute de l'Université de Californie du Sud (USC).

Par exemple, les numéros de ports standards et les numéros de protocoles sont définis par Iana.

L'attribution des numéros IP affectés aux machines connectées à l'Internet est un problème trop complexe pour être géré par un seul organisme. Iana a donc délégué à différents registres Internet (Internet Registries) des blocs de l'espace d'adressage. Ces derniers les ont parfois découpés en sous-blocs afin de les déléguer à des organismes régionaux.

Parmi les registres Internet majeurs, on peut citer RIPE pour l'Europe, APNIC pour la zone Asie-Pacifique et l'InterNIC notamment pour le continent américain.

Bien souvent, par le jeu des délégations, ce sont les fournisseurs Internet qui se voient déléguer des blocs d'adresses et qui les attribuent ainsi directement à leurs clients.

Attribution des noms de domaines

Un nom de domaine permet de regrouper des noms de machines ou de services, et se représente par une chaîne de caractères telle que `fenetre.fr` ou `fenetre.com`. Les noms de domaines ont été délégués par Iana à l'InterNIC pour les sous-domaines de `.com`, `.net`, `.org`, `.edu`, `.gov` et `.mil`. Les sous-domaines de `.fr` sont attribués par le NIC-France, dont la gestion relève de l'Inria. Des règles bien précises dans l'attribution de ces noms sont imposées par les organismes qui gèrent leurs délégations. Par exemple, la politique d'attribution de l'InterNIC est « premier arrivé, premier servi » tandis que la politique d'attribution du NIC-France est plus rigoureuse. Elle impose par exemple que les marques soient déléguées dans le sous-domaine `tm.fr`.

1.2 Les composantes de l'Internet

1.2.1 Concepts de base

Pour comprendre le fonctionnement de l'Internet, il faut successivement choisir une solution de raccordement, faire des choix pertinents quant à l'hébergement ou à la mise en place de services et résoudre les problèmes de configuration et de réglage de matériels et logiciels. Pour cela, il faut maîtriser les concepts clés de cette problématique. Attachons-nous donc à définir les termes récurrents dans le monde des réseaux : nœud, serveur, client, service, adresse IP, DNS, nom de domaine.

Nœud

L'Internet est un réseau maillé, composé d'équipements interconnectés entre eux par différents types de liaisons. On trouve principalement deux classes d'équipements : des équipements terminaux tel qu'un simple micro-ordinateur de type PC, et des routeurs, destinés à relayer les informations entre les équipements terminaux. Tous ces équipements sont qualifiés de nœuds du réseau. Les équipements terminaux possèdent une seule carte d'interface réseau, par exemple une carte Ethernet ou une carte modem. Les routeurs ayant pour charge d'interconnecter plusieurs réseaux, ils possèdent donc plusieurs cartes d'interface réseau. La figure 1.3 représente des équipements à différents niveaux de la topologie du réseau Internet.

Serveur

Le réseau maillé constitué par les routeurs de l'Internet permet à chaque nœud d'échanger des informations avec n'importe quel autre équipement quel que soit son point de rattachement. Certaines de ces machines vont héberger des logiciels répondant à des requêtes en provenance d'autres nœuds. On qualifie ces machines de serveurs, et les logiciels qu'elles hébergent sont

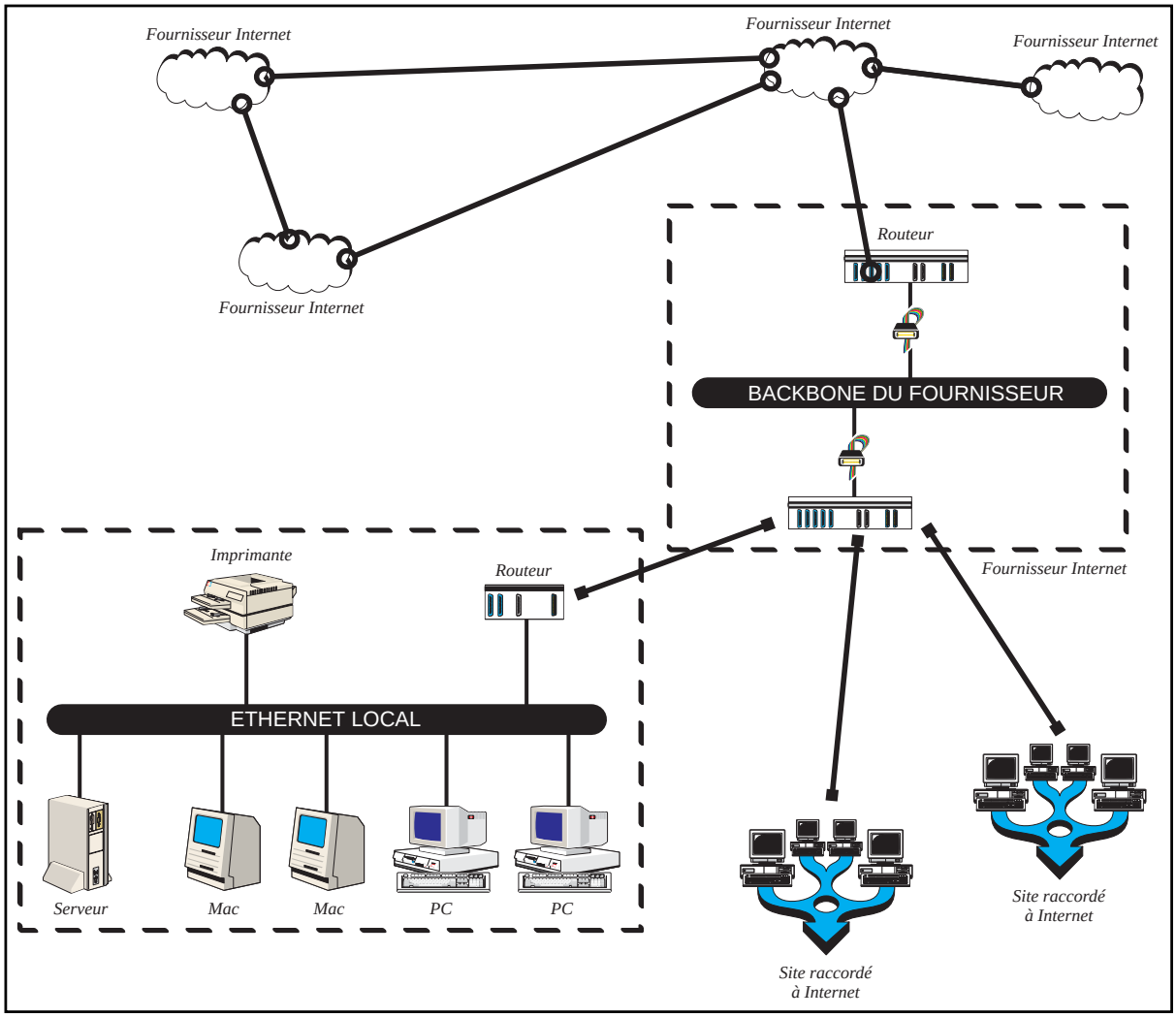

Figure 1.3 *Différents niveaux de la topologie réseau de l'Internet*

appelés logiciels serveurs. Il est d'usage d'utiliser le même terme générique *serveur* pour qualifier autant les logiciels que les machines, le contexte permettant de déterminer le sens. Un micro-ordinateur de type PC ou Macintosh peut ainsi faire office de serveur, mais lorsque la fréquence de requêtes ou la complexité de leur traitement deviennent très importantes, d'autres solutions sont mises en œuvre, souvent à l'aide de stations de travail sous le système d'exploitation Unix. Il faut savoir que certaines machines se voient proposer des centaines de requêtes par seconde.

Client

Pour pouvoir interroger un serveur, il faut disposer d'une machine connectée au réseau, ainsi que d'un logiciel adapté, appelé logiciel client. D'un côté, il propose une interface homme-machine conviviale pour vous aider à formuler une requête, et, une fois cette dernière définie,

il se connecte par le réseau au serveur distant pour dialoguer avec lui, afin de lui demander de traiter cette requête. Dès que le serveur a calculé la réponse, il la renvoie par le réseau au logiciel client, qui la présente sous une forme intelligible.

Certains serveurs et clients n'ont pas d'interaction avec l'utilisateur. Il s'agit la plupart du temps de communications client/serveur liées à des services de bas niveau, permettant par exemple à des machines de se synchroniser, ou de s'échanger des informations utiles au bon fonctionnement global du réseau. Notamment, il existe des messages de service de ce type, indiquant la présence d'un groupe de nouveaux nœuds sur le réseau ; on comprend bien que les protocoles en jeu dans ce type de scénario sont nécessaires au fonctionnement de l'Internet, mais n'ont pas de lien direct avec les utilisateurs.

Service

On définit des types de services suivant les fonctionnalités offertes par le logiciel hébergé sur un nœud. Évidemment, à chacun de ces types est associé un protocole de communication particulier permettant aux différents clients de dialoguer avec les machines qui hébergent les services auxquels ils veulent accéder. À chaque service correspondent donc des logiciels de type client, des logiciels de type serveur, et un protocole leur permettant d'échanger des informations.

Il faut noter que le développement de l'Internet a été basé sur un principe d'interopérabilité. Cela signifie que pour chaque service disponible, le protocole sous-jacent a fait l'objet d'une définition dans les moindres détails, afin de permettre à tous les clients, existants ou à venir, de collaborer avec tous les serveurs développés pour le même service, indépendamment des machines, des systèmes d'exploitation et des logiciels qui les font fonctionner. Ainsi, si on est équipé d'un micro-ordinateur de type PC sous Windows 95, rien ne nous empêche de contacter un serveur Web d'informations de météorologie hébergé sur une machine disposant d'un tout autre système d'exploitation, par exemple un serveur DEC sous OpenVMS relié à l'Internet, du moment que le logiciel client chargé sur notre matériel suit parfaitement la norme de communication du service en question. C'est là une des forces de l'Internet, dont l'accès n'est dépendant d'aucun constructeur ni éditeur de logiciel particulier, et dont les protocoles sont définis au sein de documents appelés RFC (Request for Comments) et disponibles sur le réseau lui-même. Il en existe aujourd'hui environ deux mille.

On trouve maintenant sur la plupart des ordinateurs, quel que soit le processeur dont ils disposent, des cartes d'interface permettant de les relier à l'Internet, ainsi que des logiciels client et serveur pour l'ensemble des services en vogue sur le réseau.

Adresse IP

Toute machine sur l'Internet possède un code d'identification unique, appelé adresse IP. Par exemple, le serveur Web des étudiants de l'École nationale supérieure des télécommunica-

tions de Paris possède l'adresse IP suivante : `137.194.168.13`. Aucune autre machine sur l'Internet ne possède cette même adresse. Mais nous savons qu'une machine reliée à l'Internet dispose d'une carte d'interface réseau, branchée sur son bus interne, pour la relier par un support physique (par exemple un câble Ethernet) à un routeur, lui-même connecté à l'Internet par l'intermédiaire d'autres équipements, toujours suivant le même principe. Rien n'interdit de mettre plusieurs cartes d'interfaces Ethernet dans une machine : on la transforme ainsi en routeur. C'est à chacune de ces interfaces réseau que sera associée une adresse IP unique. On comprend donc ainsi qu'une seule machine peut posséder plusieurs adresses IP.

DNS, nom de domaine

Les adresses IP sont utilisées par les logiciels clients et serveurs. Quand un utilisateur situé à Tokyo désire accéder au serveur Web des étudiants de l'ENST, il doit demander à son logiciel client de générer des paquets de données sur le réseau, qui vont être acheminés vers Paris par un ensemble de routeurs situés entre les deux villes. Afin d'être convenablement acheminés, ces paquets de données doivent tous contenir l'adresse de leur destination, c'est-à-dire `137.194.168.13`.

Mais l'utilisateur japonais qui désire accéder à ce serveur doit-il connaître cette adresse ? Heureusement non, il existe un annuaire distribué sur l'Internet, nommé DNS, dont l'objet est de fournir des correspondances entre des noms de machines et des adresses IP. Il simplifie la vie des utilisateurs : il est plus simple de se souvenir d'un nom que d'une adresse IP, d'autant plus que l'adresse d'une machine dépend du réseau auquel elle est rattachée, et elle vient donc à changer quand la machine est déménagée.

Ainsi, l'utilisateur japonais indique simplement à son logiciel qu'il désire accéder à la machine dont le nom est `www-stud.enst.fr`. Le logiciel en question se contente d'interroger une machine gérant le service DNS, afin d'être informé de la correspondance entre `www-stud.enst.fr` et `137.194.168.13`. Ensuite, il est à même de générer les paquets de données à destination de cet équipement. Évidemment, l'opération inverse est tout aussi possible : on peut fournir à un serveur DNS une adresse IP afin d'en obtenir un nom de machine.

La machine `www-stud.enst.fr` appartient au domaine `enst.fr`. Ce domaine contient des sous-domaines, par exemple `res.enst.fr`, qui contiennent à leur tour des noms de machines, par exemple `pcahmed.res.enst.fr`. Dans les RFC, qui sont écrits en anglais, on trouve l'expression « *domain name* » pour désigner autant les noms de machines que les noms de domaine. En effet, un « *domain name* » signifie tantôt un *nom de domaine*, tantôt un *nom dans le domaine*.

Nous avons remarqué précédemment qu'une machine qui dispose de plusieurs cartes d'interface réseau dispose du coup de plusieurs adresses IP. Sachant qu'un nom de domaine est associé à chaque adresse IP, on remarquera qu'une machine peut ainsi posséder plusieurs noms (autant que d'interfaces réseau), comme le montre la figure 1.4 page suivante.

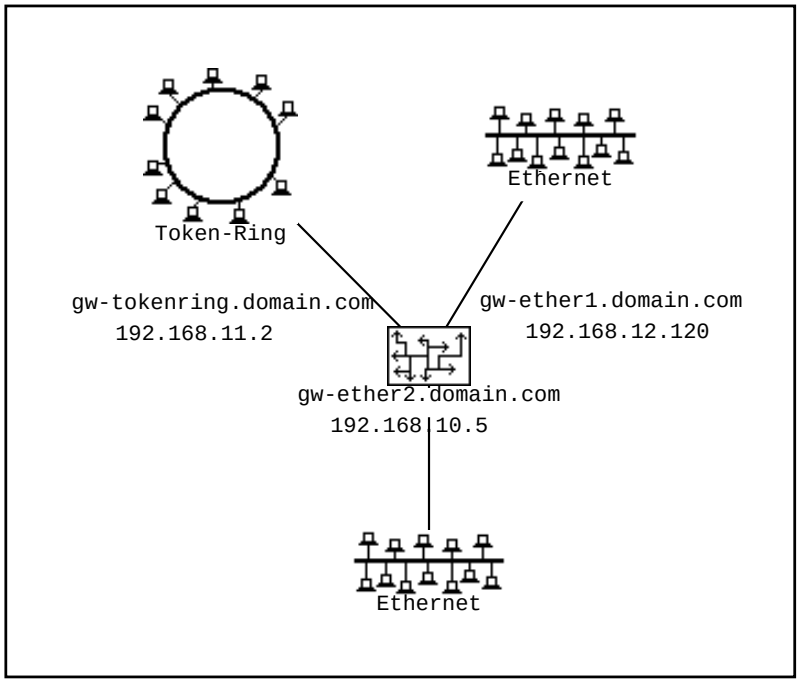

Figure 1.4 *Différents noms de domaines et adresses IP pour un même routeur*

1.2.2 Aperçu des services disponibles sur l'Internet

On classe généralement les services disponibles sur l'Internet en deux catégories : les services de base, quasiment nécessaires à tout organisme connecté au réseau, et les services d'information, souvent apparus relativement tard sur le réseau.

Les services de base

Les services de base comprennent la messagerie, le DNS, les forums et l'accès distant.

La messagerie (courrier électronique)
Elle permet aux utilisateurs d'échanger entre eux des messages composés d'un texte et éventuellement de pièces jointes, par exemple un fichier établi avec un tableur ou un traitement de texte. Souvent ces messages ne comportent que quelques lignes. L'envoi d'un message de ce type nécessite un logiciel adéquat sur le poste de travail, auquel on doit fournir l'adresse du destinataire. De même qu'une adresse postale, l'adresse électronique du destinataire est unique, et est souvent du type `Prenom.Nom@domaine.fr` (exemple : `Luc.Stoned@fenetre.fr`). Le transport d'un message à travers plusieurs continents entre l'émetteur et le destinataire prend tout au plus quelques dizaines de secondes, dans la plupart des cas.

Le DNS
Nous en avons déjà dit quelques mots auparavant, c'est un service réparti dans lequel parti-

cipent de nombreuses machines sur l'Internet. Il permet d'associer des noms de machines à des adresses IP et réciproquement. Nous verrons par la suite qu'il fournit aussi de nombreux autres services, notamment dans le procédé d'acheminement du courrier électronique.

Les forums
Les forums, aussi appelés News, regroupent au sein d'un même service des lieux d'échange où les utilisateurs peuvent dialoguer. Ce service n'est pas disponible uniquement sur des machines connectées à l'Internet : on désigne par Usenet le réseau formé par toutes les machines qui peuvent recevoir les forums. Le principe est simple : on dispose un serveur News sur le site ; ce dernier est relié à d'autres serveurs du même type, par exemple par l'intermédiaire de l'accès Internet ; les connexions entre serveurs News forment ainsi un maillage sur toute la planète, et chacun communique avec ses quelques voisins. Chaque message envoyé sur un quelconque serveur News est ainsi distribué par ce dernier à ses voisins, qui à leur tour le redistribuent ; il s'agit ici de la technique de propagation par inondation. Ainsi, en quelques heures, tout au plus un ou deux jours, ce message aura été distribué à l'ensemble des serveurs News du réseau. Et bien sûr, on peut consulter sur notre serveur l'ensemble de ces messages, au nombre de plus d'une centaine de milliers diffusés chaque jour. Ce service s'appelle forums car les messages y sont classés parmi plus de 8000 thèmes précis.

L'accès distant
L'accès à des ordinateurs distants par le service `telnet` permet de contrôler un matériel éloigné comme on le ferait avec un terminal connecté sur son port console.

Les services d'information

Les services d'information comprennent le World Wide Web, le transfert de fichiers, WAIS, IRC et les services IP multicast.

Le World Wide Web
le World Wide Web est le service le plus utilisé en terme de nombre de paquets qui transitent sur le réseau. C'est souvent le premier service auquel sont confrontés les utilisateurs de l'Internet. Il permet d'accéder à des documents multimédias sur des serveurs WWW. La caractéristique principale de ce service est qu'un serveur peut à tout instant rediriger l'utilisateur vers des informations contenues sur un autre serveur WWW. Ainsi, l'utilisateur du WWW se balade de serveur en serveur, au fil de son utilisation ; l'expression en vogue pour ce type d'activité est 'Surfer sur le Web'. De par le caractère générique de ce service (tout document multimédia peut être mis en accès par un serveur Web : son, image, réalité virtuelle, etc.), de très nombreux services Web ont été développés. On peut par exemple commander une pizza par le Web, ou bien observer le contenu d'une pièce par l'œil d'une caméra située à des milliers de kilomètres de son poste de travail.

Le transfert de fichiers
Le transfert de fichiers, connu sous le nom de FTP, permet de rapatrier des fichiers quelconques, par exemple des logiciels, depuis des serveurs d'archive. On trouve sur l'Internet un

ensemble de sites ouverts à tous et archivant des dizaines de milliers de fichiers. D'autres serveurs, appelés « *sites miroirs* » recopient régulièrement le contenu de ces serveurs d'archive. Les utilisateurs se connectent ainsi sur le miroir le plus proche afin de récupérer les fichiers qui les intéressent.

Les bases de données WAIS
Les serveurs WAIS permettent de rechercher des informations sur d'immenses bases de données. La bibliothèque de l'Inria fournit par exemple un accès à la base de données des ouvrages dont elle dispose, sur un serveur WAIS.

Le dialogue interactif : IRC
le service IRC (Internet Relay Chat) permet de dialoguer dans des forums, mais à la différence des News, le dialogue est ici en direct : tout le monde voit immédiatement les textes saisis sur la console, et réciproquement, on voit s'afficher à l'écran ce que tous les autres utilisateurs du service IRC décident de frapper sur leur clavier. L'interactivité est donc ici totale.

Les services IP multicast
les services IP multicast permettent l'audio et la vidéo-conférence, ainsi que de nombreux autres services coopératifs. L'ensemble des machines de l'Internet qui y ont accès s'appelle le MBone. Par exemple, la Nasa retransmet systématiquement, en audio et vidéo sur le réseau MBone, les phases importantes des missions de la navette spatiale américaine.

1.2.3 L'Internet en chiffres

L'Internet est le plus grand réseau décentralisé d'ordinateurs au monde. De nombreuses études ont été menées dans le but de faire l'inventaire des ressources qui le composent. Ainsi, on dispose de chiffres assez précis renseignant sur son nombre de nœuds et sur les services qui y sont offerts.

On comptabilise actuellement 9,5 millions de machines connectées à l'Internet, dont 2 millions en Europe et 140 000 en France.

Chaque jour, entre 100 000 et 150 000 articles sont postés dans les quelques 8 000 forums ; cela correspond à un volume de plus de 400 Mo, et un débit de plus de 75 articles par minute.

Le réseau IRC est composé de plus de 130 serveurs, et on y trouve environ 3 500 forums de discussion interactive. Il y a souvent plus de 6 000 utilisateurs connectés simultanément sur l'ensemble de ces forums.

On dénombre plus de 300 000 serveurs Web. Le moteur de recherche sur l'Internet de DEC, AltaVista, a recensé jusqu'à maintenant plus de 30 millions de pages Web.

Il faut bien noter que ces chiffres sont en constante évolution et qu'il est toujours difficile d'estimer une grandeur sur l'Internet, qu'on parle de nombre de machines ou de trafic. Une constante se maintient néanmoins : le doublement de ces chiffres chaque année.

2

Notions de base des réseaux TCP/IP

L'Internet est fondé sur la famille des protocoles TCP/IP, qui permettent aux machines de communiquer à travers un réseau local (LAN, Local Area Network) ou longue-distance (WAN, Wide Area Network). Les protocoles TCP/IP peuvent être utilisés de façon interne à une société, on parle alors d'intranet, ou bien pour se connecter au réseau Internet.

2.1 Les différentes classes d'adresses

L'Internet est un réseau basé sur un ensemble de protocoles : les protocoles de la famille TCP/IP. La version actuelle est nommée IPv4 (version 4). Une nouvelle version, proposant de très profondes modifications vient d'être adoptée : IPv6 ou IP-NG (IP Next Generation). Nous allons nous attacher à décrire IPv4, IPv6 n'étant qu'à un stade d'expérimentation, sur l'Internet.

Pour localiser les machines, on fait usage d'adresses. Ces dernières sont utilisées à de nombreux niveaux dans les paquets qui transitent sur le réseau. Les adresses IP sont de la forme suivante : `x.y.z.t` où x, y, z et t sont des nombres compris entre 0 et 255 (octets). Par exemple, `137.194.168.13` est l'adresse IP d'une machine connectée sur l'Internet. Il faut exactement huit chiffres binaires pour coder chacun des quatre octets constituant une adresse IP. On peut par exemple remarquer que 0 se code `00000000`, 255 se code `11111111` et 137 se code `10001001`. Les adresses IP contenant 4 nombres codés chacun sur huit chiffres binaires, on peut les représenter sur 32 bits. Ces 32 bits sont séparés en deux zones de bits contiguës : une partie décrit le numéro de réseau auquel est rattaché l'équipement, et une

partie informe du numéro de l'équipement lui-même, appelée numéro d'hôte. Dans cette dernière partie, deux numéros sont réservés : celui où tous les bits sont nuls (on indique ainsi le réseau lui-même, une adresse de ce type s'appelle adresse réseau), et celui où tous les bits sont à 1 (on indique alors l'ensemble des machines, une adresse de ce type s'appelle adresse de diffusion, ou adresse *broadcast*). Une adresse est réservée pour indiquer toutes les machines connectées sur un support physique donné : `255.255.255.255` (certains systèmes d'exploitation utilisent `0.0.0.0` au lieu de `255.255.255.255`, mais ils sont de moins en moins répandus).

Il existe quatre classes d'adresses avec la version 4 (version courante) des protocoles TCP/IP, car les parties réseau et hôte n'ont pas toujours la même taille. Détaillons donc l'espace d'adressage d'IPv4.

2.1.1 Les adresses de classe A

Les adresses de classe A ont une partie réseau sur 8 bits, et une partie hôte sur 24 bits. Leur bit de poids le plus fort est 0, ce qui permet de les distinguer des autres classes.

Leur forme est décrite par la figure 2.1.

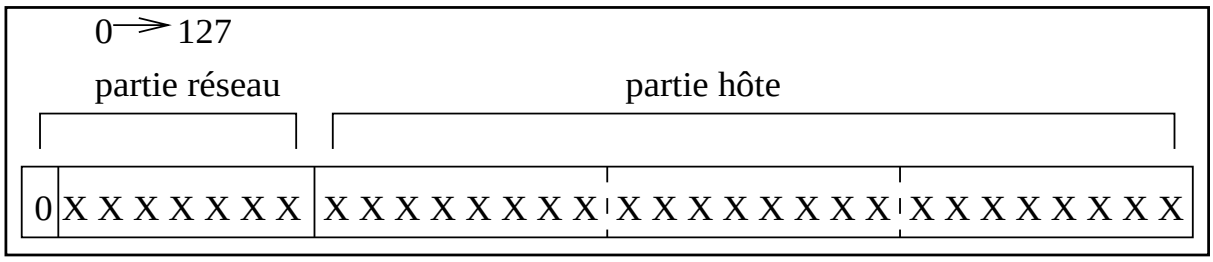

Figure 2.1 *Adresses de classe A*

Elles commencent en `0.0.0.0` et se terminent en `127.255.255.255`. Il y a donc 128 réseaux de classe A, chacun pouvant accueillir théoriquement jusqu'à $2^{24} - 2$ hôtes (l'adresse réseau et l'adresse de diffusion ne désignent pas d'hôte particulier). Ces 127 adresses sont déjà toutes réservées sur l'Internet, un site ne peut donc pas en obtenir, mis à part le réseau de classe A d'adresse 127 qui est un réseau fictif interne à chaque machine : chaque nœud s'identifie au hôte 1 de ce réseau, dont l'adresse IP est `127.0.0.1`, et qu'on appelle `localhost`. Pour comprendre cela, rappelons-nous qu'à chaque interface physique d'une machine, une carte Ethernet par exemple, est associée une adresse IP unique. Souvent, le système d'exploitation crée une interface virtuelle supplémentaire, appelée « pseudo-interface loopback », et lui attribue l'adresse `127.0.0.1`. Ainsi, un programme cherchant à s'adresser, à l'aide des protocoles TCP/IP, à un autre programme situé sur la même machine n'a qu'à indiquer `127.0.0.1` dans le champ destination des datagrammes IP qu'il génère. Ces datagrammes ne quitteront pas la machine, car le système d'exploitation reconnaîtra alors cette adresse comme étant celle de son interface *loopback*.

2.1.2 Les adresses de classe B

Les adresses de classe B ont une partie réseau sur 16 bits, et une partie hôte de même taille. Leurs deux bits de poids fort sont 10, ce qui permet de les distinguer des autres classes.

Leur forme est décrite par la figure 2.2.

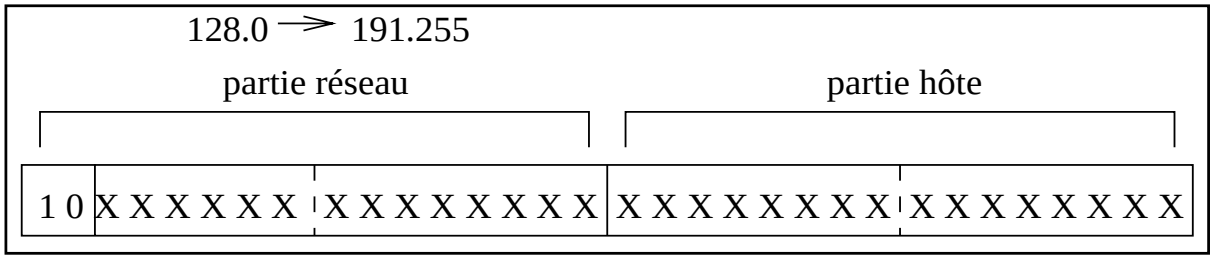

Figure 2.2 *Adresses de classe B*

Elles commencent en `128.0.0.0` et se terminent en `191.255.255.255`. Il y a donc 16 384 réseaux de classe B, chacun pouvant accueillir jusqu'à 65 534 hôtes. Une grande partie de ces 16 384 classes est déjà réservée. Pour en obtenir, il faut justifier qu'on s'apprête à connecter un réseau de très grande envergure à l'Internet.

2.1.3 Les adresses de classe C

Les adresses de classe C ont une partie réseau sur 24 bits, et une partie hôte sur 8 bits. Leurs trois bits de poids fort sont 110, ce qui permet de les distinguer des autres classes.

Leur forme est décrite par la figure 2.3.

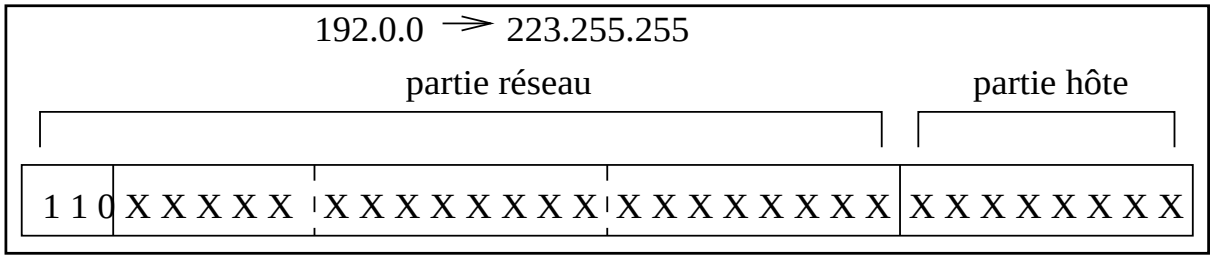

Figure 2.3 *Adresses de classe C*

Elles commencent en `192.0.0.0` et se terminent en `223.255.255.255`. Il y a donc 2 097 152 réseaux de classe C, chacun pouvant accueillir jusqu'à 254 hôtes. Il reste encore suffisamment de classes C pour pouvoir en distribuer encore pendant entre cinq et dix ans, d'après de récentes analyses fondées sur le taux de croissance estimé de l'Internet.

2.1.4 Les adresses de classe D

Leur forme est décrite par la figure 2.4.

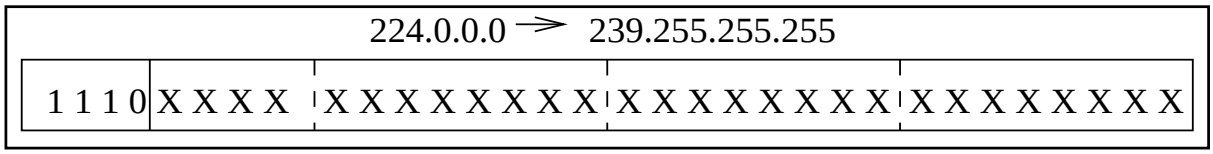

Figure 2.4 *Adresses de classe D*

On les appelle aussi adresses de groupes multicast. Elles commencent en `224.0.0.0` et se terminent en `239.255.255.255`. Ce sont des adresses particulières où la notion de réseau disparaît : elles désignent non pas un hôte particulier, mais un groupe d'hôtes. Tout équipement désirant faire partie d'un groupe peut demander à y adhérer en précisant l'adresse multicast correspondante. À tout moment, tout paquet émis par une machine quelconque sur l'Internet, et à destination d'une adresse multicast particulière, est acheminé vers tous les membres du groupe en question.

Certaines adresses de groupe sont déjà attribuées, en voici un échantillon :

Adresse	Nom associé	Description
`224.0.0.0`	BASE-ADDRESS.MCAST.NET	début des adresses multicast
`224.0.0.1`	ALL-SYSTEMS.MCAST.NET	toutes les machines
`224.0.0.2`	ALL-ROUTERS.MCAST.NET	tous les routeurs
`224.0.0.12`	DHCP-AGENTS.MCAST.NET	agents DHCP
`224.0.0.13`	PIM-ROUTERS.MCAST.NET	routeurs multicast supportant PIM

2.1.5 Les adresses de classe E

Les adresses de classe E débutent en `240.0.0.0` et se terminent en `255.255.255.255`. Elles sont réservées par Iana. Seule `255.255.255.255` est pour l'instant attribuée ; elle désigne toutes les machines, et est utilisée lorsqu'on a besoin de s'adresser à tous les équipements directement connectés à un même support : un paquet à destination de cette adresse ne traverse jamais les routeurs.

Leur forme est décrite par la figure 2.5.

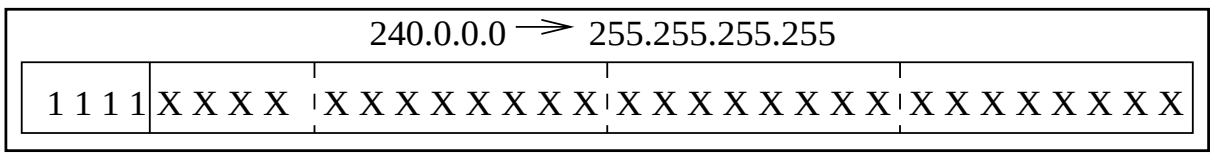

Figure 2.5 *Adresses de classe E*

2.2 Les masques de réseau et de sous-réseaux

2.2.1 Les masques de réseau

À chaque classe d'adresses est associé un masque de réseau, ou *netmask*, qui est constitué de 32 bits. Le tableau 2.1 fournit les différents masques pour les trois classes traditionnelles.

classe	masque
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

Tableau 2.1 *Masques de réseau*

Un `et` logique appliqué entre le masque de réseau et une adresse IP permet d'obtenir l'adresse du réseau correspondant.

Un `et` logique appliqué entre le complément[1] à 1 du masque de réseau et une adresse IP permet d'obtenir la partie hôte correspondante. Ainsi, à l'aide du masque de réseau, on peut définir, pour toute adresse IP :

- l'adresse réseau associée ;
- la partie hôte associée ;
- l'adresse de diffusion associée qui désigne tous les hôtes de ce réseau.

Le tableau 2.2 fournit ces informations pour trois adresses IP prises parmi les trois classes fondamentales.

adresse IP	10.25.2.5	172.17.5.8	192.168.53.24
classe	A	B	C
masque de réseau	255.0.0.0	255.255.0.0	255.255.255.0
adresse réseau	10.0.0.0	172.17.0.0	192.168.53.0
adresse de diffusion	10.255.255.255	172.17.255.255	192.168.53.255

Tableau 2.2 *Données associées à une adresse IP*

2.2.2 Les masques de sous-réseaux

Parfois, on est amené à répartir les adresses IP d'un même réseau de classe A, B ou C sur plusieurs supports physiques. En effet, si on dispose d'une cinquantaine de machines, à répartir

1. Pour complémenter à 1 un nombre, il suffit de l'écrire en binaire et de remplacer ses bits à 1 en bits à 0 et réciproquement.

sur trois réseaux Ethernet par exemple, notre fournisseur Internet ne va pas nous offrir trois réseaux de classe C : une seule classe C peut déjà accueillir 254 machines.

Pour résoudre ce problème, il faut introduire un nouveau type de masque : le masque de sous-réseaux.

Le principe est simple : le réseau est découpé en sous-réseaux de même taille. Pour cela, la partie hôte des adresses est elle-même découpée en deux plages de bits : la plage correspondant aux bits positionnés à 1 dans le masque de sous-réseaux désigne le numéro du sous-réseau, et l'autre plage désigne le numéro de machine.

Pour trouver le sous-réseau auquel appartient un équipement, il suffit donc d'appliquer un `et` logique entre son adresse IP et le masque de sous-réseaux.

Parmi les numéros de sous-réseaux ainsi créés, deux sont interdits d'utilisation : le sous-réseau 0 et le sous-réseau où tous les bits sont à 1. On perd donc d'entrée de jeu deux adresses de sous-réseaux. De plus, l'ensemble des sous-réseaux répartis sur un site doit être connexe. C'est-à-dire qu'on ne peut pas, par exemple, attribuer deux sous-réseaux d'une même classe C sur deux câbles Ethernet reliés par un anneau FDDI d'une autre classe C.

Découpage en sous-réseaux: application pratique

Appliquons ce principe à un exemple simple. Supposons que nous possédions une classe C (`192.168.22.0`) et un réseau local constitué de trois brins Ethernet. Le premier relie deux serveurs ainsi qu'un routeur Internet, le second relie deux Macintosh, et le troisième relie trois PC, comme indiqué sur la figure 2.6 page suivante. Il nous faut donc découper notre classe C au moins en trois sous-réseaux. Sachant que deux sous-réseaux sont réservés, il en faut donc au moins 5. Pour coder 5 sous-réseaux, il faut trois bits, car $2^2 \leq 5 < 2^3$. Or nous disposons d'une classe C dont le masque de réseau s'écrit ainsi en binaire :
`11111111.11111111.11111111.00000000`.
Il nous faut donc réserver trois bits dans la partie hôte, c'est-à-dire dans le dernier octet. Le masque de sous-réseaux correspondant s'écrit donc comme suit :
`11111111.11111111.11111111.11100000`.
En décimal, cela donne : `255.255.255.224`.
Nous avons pris trois bits pour définir le sous-réseau, il y a donc huit sous-réseaux qu'on peut récapituler dans le tableau 2.3 page 40.

2.3 L'attribution des adresses IP

Comme toutes les ressources numériques de l'Internet, l'espace d'adressage IPv4 est géré par Iana. Dans un but de délocalisation de la tâche d'attribution des classes d'adresses IP, Iana a délégué les plages d'adresses à différents organismes. Pour obtenir une classe d'adresses, il faut contacter l'organisme qui fait géographiquement compétence. Le plan d'adressage actuel

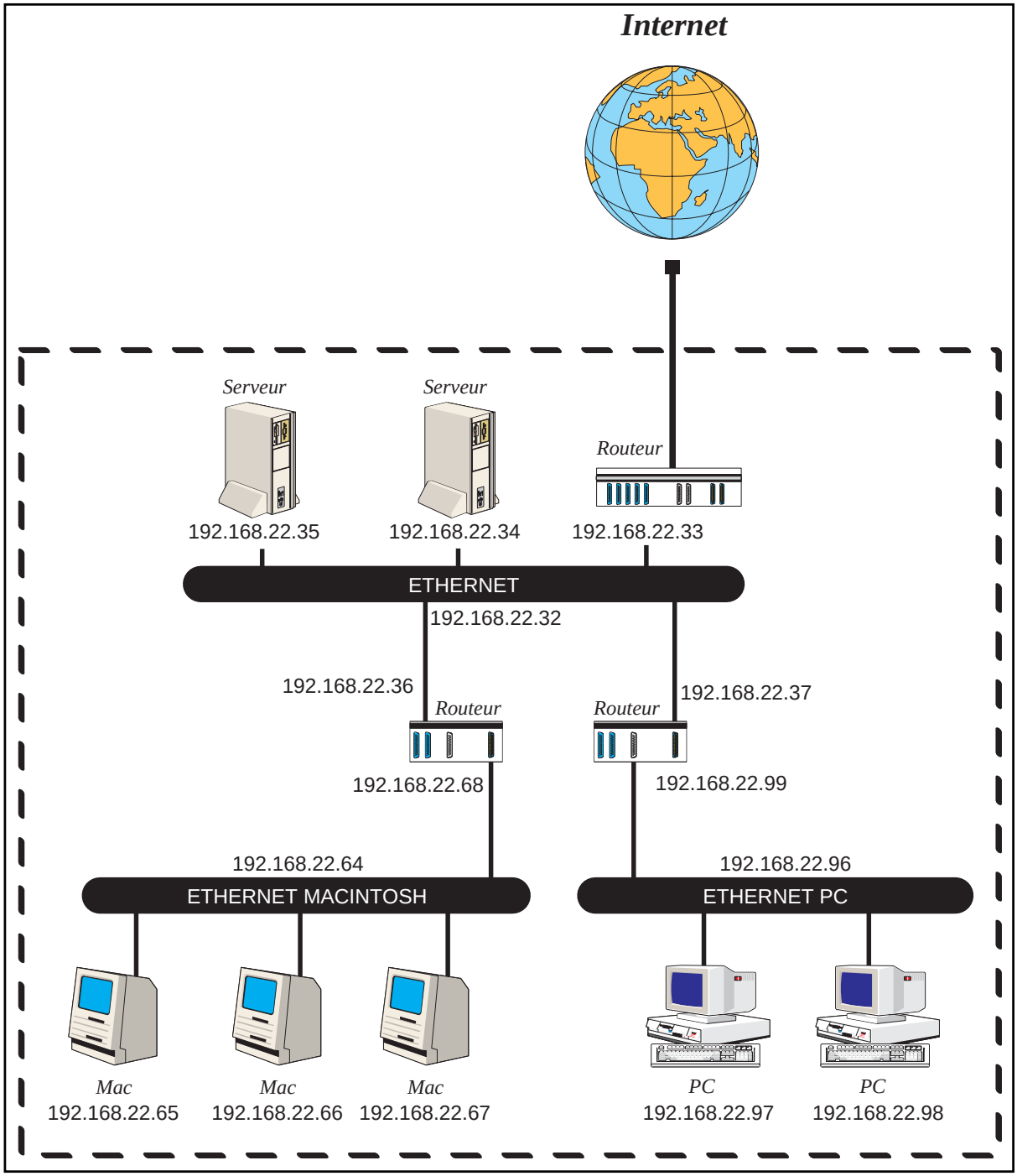

Figure 2.6 *Plan d'adressage de sous-réseaux*

numéro	adresse du sous-réseau	masque de sous-réseau	adresse de diffusion	plage	fonction
0	192.168.22.0	255.255.255.224	192.168.22.31	192.168.22.0 à 192.168.22.31	réservé
1	192.168.22.32	255.255.255.224	192.168.22.63	192.168.22.32 à 192.168.22.63	serveurs
2	192.168.22.64	255.255.255.224	192.168.22.95	192.168.22.64 à 192.168.22.95	Macintosh
3	192.168.22.96	255.255.255.224	192.168.22.127	192.168.22.96 à 192.168.22.127	PC
4	192.168.22.128	255.255.255.224	192.168.22.159	192.168.22.128 à 192.168.22.159	inutilisé
5	192.168.22.160	255.255.255.224	192.168.22.191	192.168.22.160 à 192.168.22.191	inutilisé
6	192.168.22.192	255.255.255.224	192.168.22.223	192.168.22.192 à 192.168.22.223	inutilisé
7	192.168.22.224	255.255.255.224	192.168.22.255	192.168.22.224 à 192.168.22.255	réservé

Tableau 2.3 *Masques de sous-réseau*

est décrit dans le tableau suivant, les blocs d'adresses étant notés au format CIDR[2] :

Bloc d'adresses (format CIDR)	Registre obtenant la délégation	Date de délégation
000-063/8	Iana	Septembre 81
064-095/8	Iana - Réservé	Septembre 81
096-126/8	Iana - Réservé	Septembre 81
127/8	Iana	Septembre 81
128-191/8	différents registres	Mai 93
192-193/8	différents registres	Mai 93
194-195/8	RIPE NCC - Europe	Mai 93
196-197/8	InterNIC	Mai 93
198-199/8	InterNIC - Amérique Centrale et du Sud	Mai 93
200-201/8	APNIC - Pacifique	Mai 93
204-205/8	InterNIC - Amérique du Nord	Mars 94
206/8	InterNIC - Amérique du Nord	Avril 95
207/8	InterNIC - Amérique du Nord	Novembre 95
208/8	InterNIC - Amérique du Nord	Avril 96
209/8	InterNIC - Amérique du Nord	Juin 96
210/8	APNIC - Pacifique	Juin 96
211/8	APNIC - Pacifique	Juin 96
212-223/8	Iana - Réservé	Septembre 81
224-239/8	Iana - Multicast	Septembre 81
240-255/8	Iana - Réservé	Septembre 81

2. Ce format est décrit section 2.6.4 page 53 ; précisons ici simplement que, par exemple, 206/8 désigne les adresses comprises entre `206.0.0.0` et `206.255.255.255`.

RIPE (Réseaux IP Européens) est l'organisme compétent en Europe. APNIC gère la zone Asie-Pacifique. L'InterNIC gère de nombreuses zones géographiques, notamment l'Amérique, et il existe plusieurs autres organismes auxquels Iana a délégué une partie de l'espace d'adressage. Ces différents registres Internet délèguent parfois à leur tour à d'autres entités locales. Par exemple, le NIC-France (géré par l'Inria) s'est occupé pendant une période d'attribuer des classes d'adresses aux organismes désireux de se connecter à l'Internet depuis la France.

Attribution d'un réseau de classe A, B ou C

En France, ce sont les fournisseurs qui gèrent ce service, chacun disposant d'un ensemble de classes C qui lui ont été déléguées par RIPE, ou par le NIC-France. Ainsi, c'est au fournisseur d'attribuer une ou plusieurs classes C. Celui qui désire une classe B doit s'adresser directement à RIPE ou à l'InterNIC, et fournir un dossier justifiant sa demande. Personne ne peut obtenir de classe A, elles ont toutes déjà été déléguées.
Lorsqu'un client utilisant des adresses attribuées par un fournisseur décide de résilier son abonnement pour se connecter chez un concurrent, il doit rendre ces adresses et renuméroter son réseau avec de nouvelles adresses fournies par le nouveau fournisseur. Cette opération est parfois complexe et coûteuse mais est imposée pour des raisons techniques, notamment pour juguler l'explosion des tables de routage des routeurs des fournisseurs du monde entier.

2.4 Les couches de protocoles

2.4.1 Le modèle OSI

L'ISO (International Standard Organisation), organisation de normalisation internationale, a développé un modèle d'interconnexion de systèmes ouverts (Open Systems Interconnection) appelé modèle OSI. Dans ce modèle de référence, les protocoles réseau sont organisés en couches, chacune utilisant les services de celle juste au-dessous, et fournissant des services à celle juste au-dessus. L'OSI a formalisé sept couches, représentées sur la figure 2.7 page suivante.

La couche *physique* définit les fonctions physiques de la liaison, par exemple les caractéristiques électriques, mécaniques et/ou optiques du support utilisé.

La couche *liaison de données* définit la manière dont les informations sont échangées entre deux matériels directement connectés par un même support physique. Au niveau de cette couche, les données sont rassemblées en trames. Des fonctions de contrôle d'erreur et de flux peuvent être éventuellement apportées à ce niveau, et dans le cas où le support physique est partagé entre plusieurs machines, un contrôle d'accès au support est présent. C'est notamment ce MAC (Medium Access Control) qui caractérise le réseau local.

La couche *réseau* fournit la fonction d'adressage et de routage, les paquets qu'elle traite sont encapsulés dans les trames du niveau 2, et sont acheminés d'un bout à l'autre du réseau en

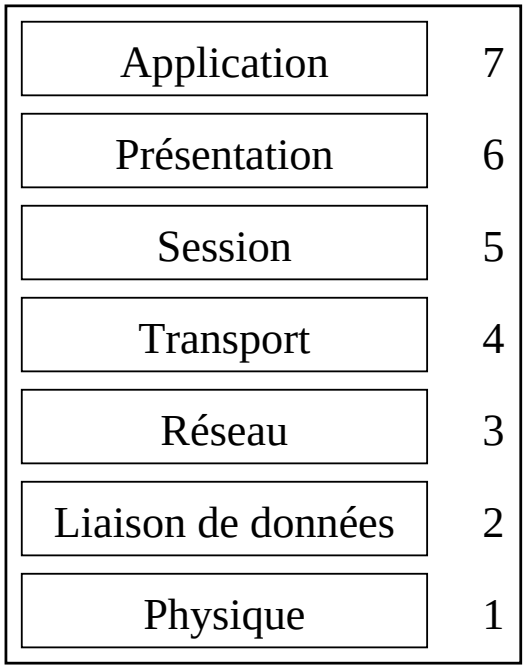

Figure 2.7 *Les sept couches OSI*

traversant les routeurs.

La couche *transport* effectue des contrôles supplémentaires à la couche réseau, et fournit ainsi plusieurs types de qualité de service distincts, au choix de l'utilisateur de ce niveau 4.

La couche *session* introduit la notion d'établissement de sessions, service destiné à structurer et synchroniser les données échangées entre les applications.

La couche *présentation* permet l'interopérabilité de systèmes différents, dialoguant à travers le réseau, en normalisant les types des données pouvant être échangées ainsi que leur codage.

La couche *application* intègre les logiciels qui utilisent les ressources du réseau.

2.4.2 TCP/IP

Les protocoles de la famille TCP/IP, utilisés sur l'Internet, correspondent principalement aux niveaux 3 et 4 du modèle OSI, et sont constituées selon le modèle de la commutation de paquets, à opposer au modèle de la commutation de circuits, utilisé par exemple sur le réseau téléphonique commuté. Les protocoles TCP/IP commencent au niveau de la couche réseau ; les couches inférieures sont capables de transporter différentes familles de protocoles, ce qui permet, sur un même réseau physique, à des machines de communiquer par TCP/IP, et aussi, sur le même support, par d'autres protocoles comme DECnet.

Chaque couche vient ajouter un en-tête aux données que lui fournit la couche supérieure. Les informations rajoutées sont caractéristiques des services de la couche correspondante.

Le tableau 2.4 présente les quatre couches les plus basses du modèle. La figure 2.8 page suivante montre une capture d'écran du logiciel Packetman[3] d'analyse de trafic, sur laquelle on peut identifier les datagrammes ayant transité sur un réseau local et l'encapsulation en couche des données qu'ils transportent.

Couches	**Protocoles**				
Transport	TCP	UDP	ICMP		
Réseau	IP			ARP	
Liaison de données	LLC 802.2				
	FDDI		802.3	802.4	802.5
Physique	Ethernet, paire torsadée, etc...				

Tableau 2.4 *Encapsulation en couches*

2.4.3 La couche physique

Les couches physiques permettent de s'adapter à différents supports. Par exemple, un câble coaxial, une fibre optique ou une onde radio nécessitent des couches physiques distinctes.

Le câble coaxial de type Ethernet constitue le support physique le plus utilisé sur les réseaux locaux.

2.4.4 La couche liaison de données

La couche liaison de données est souvent découpée en deux sous-couches distinctes : la sous-couche LLC (Logical Link Control) supérieure (IEEE 802.2) est commune pour toutes les sous-couches MAC (Medium Access Control) inférieures, que sont FDDI (ISO 9314-2), IEEE 802.3 CSMA/CD très proche de l'Ethernet (architecture en bus à détection de collision), IEEE 802.4 (bus à jeton) et IEEE 802.5 (anneau à jeton).

Cette couche doit contrôler l'accès au support, et permettre l'échange de trames entre les machines *directement* connectées sur un même support physique. Pour cela, l'en-tête des messages émis par la couche liaison de données contient les adresses de la machine émettrice (adresse source), de la machine réceptrice (adresse destination), et le type de paquet encapsulé (IP, ARP, autre). Dans le cas d'un réseau local Ethernet, il s'agira d'adresses Ethernet à six octets. L'adresse destination Ethernet d'une trame désigne une machine directement connectée par le même support physique, le prochain routeur à qui la trame est destinée, par exemple. Dans la cadre d'Ethernet, il existe une adresse de diffusion particulière qui permet d'émettre une trame à destination de toutes les machines connectées sur le support : `FF:FF:FF:FF:FF:FF`.

3. Packetman est un logiciel développé par l'université de Curtin, dont on peut trouver les binaires sur l'Internet.

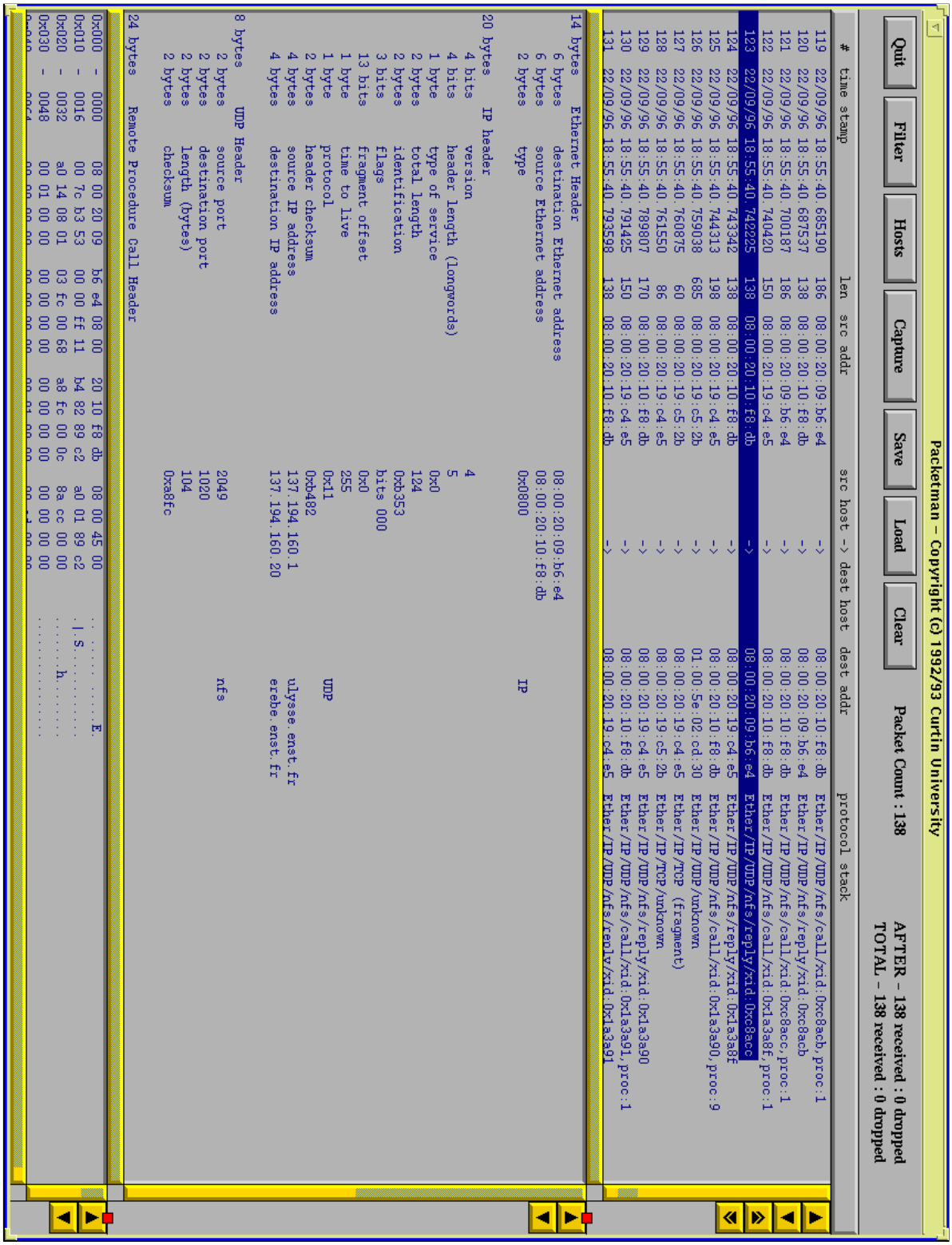

Figure 2.8 *Le logiciel Packetman*

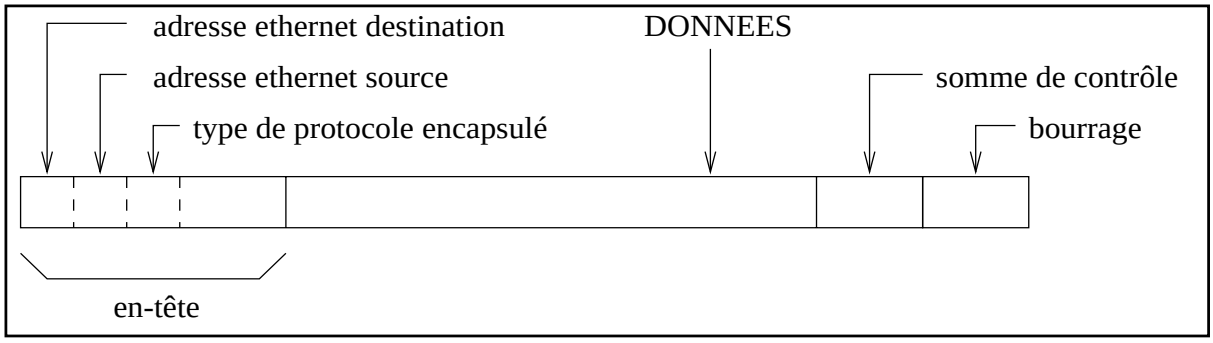

Figure 2.9 *Exemple de couche liaison de données : la trame Ethernet*

2.4.5 La couche réseau

La couche réseau est la première couche faisant réellement partie de la famille des protocoles TCP/IP. On y trouve principalement le protocole IP (Internet Protocol) et, sur certains supports partagés par plusieurs équipements, comme Ethernet, le protocole ARP (Address Resolution Protocol).

Le protocole IP

Le protocole IP est la brique de base qui permet d'émettre des paquets d'informations à travers le réseau. Les données des autres protocoles de l'Internet sont encapsulées dans des paquets IP.

Les principaux champs de l'en-tête IP sont :

- l'adresse IP source ;
- l'adresse IP destination ;
- le type de paquet encapsulé dans la partie données.

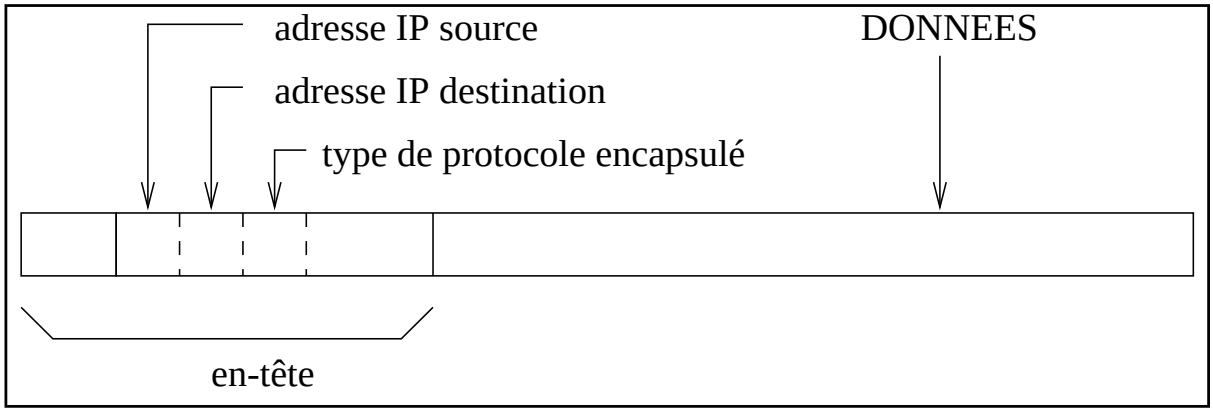

Figure 2.10 *Le datagramme IP*

Le protocole ARP

Lorsqu'un équipement A désire communiquer avec un équipement B qui est connecté sur le même support, pour lui fournir un datagramme IP, celui-ci doit être encapsulé dans une trame de la couche MAC dont l'en-tête contient l'adresse Ethernet de B. Mais l'équipement A ne connaît, *a priori*, que l'adresse IP de B. Il faut donc un protocole qui permette à une machine quelconque d'un support partagé de connaître les adresses de couche liaison de données des autres équipements à l'aide de la simple connaissance des adresses de couche réseau. C'est l'objectif du protocole ARP :

- la machine A connaît l'adresse IP de B et veut lui faire parvenir un datagramme IP ;
- A diffuse donc une trame MAC de diffusion comportant dans la partie données l'adresse IP de B ;
- toutes les machines branchées sur le support décodent cette trame et comparent l'adresse IP contenue à la leur ;
- B reconnaît ainsi son adresse IP et renvoie une trame ARP à A pour lui indiquer son adresse MAC.
- A connaît ainsi l'adresse Ethernet de B et peut donc construire l'en-tête d'une trame MAC encapsulant le datagramme IP à destination de B.

2.4.6 La couche transport

Les principaux protocoles de la couche transport sont au nombre de trois : TCP (Transmission Control Protocol) fournit un service fiable d'échange de flot continu de données, appelé mode connecté, UDP (User Datagram Protocol) fournit un service d'échange de datagrammes, appelé mode non connecté, et ICMP (Internet Control Message Protocol) permet de transporter des informations de service, comme par exemple la signalisation de l'inaccessibilité d'un nœud du réseau, ou une demande de redirection émise par un routeur vers un hôte.

Plusieurs services sont construits sur les protocoles TCP et UDP, c'est pourquoi on trouve dans les en-têtes des paquets qu'ils génèrent un numéro de port qui permet à la machine réceptrice de renvoyer ces données au processus gérant le service correspondant.

Un certain nombre de ports sont normalisés, le tableau 2.5 page suivante dresse la liste des plus importants.

2.4.7 Les couches supérieures

Les couches au-dessus du niveau transport proposent des services de toutes sortes, en utilisant soit TCP, soit UDP. Parmi ces services, on trouve la messagerie (POP pour la consultation, SMTP pour l'envoi), l'échange de fichiers (FTP), les services d'information (WWW, Gopher, WAIS), etc.

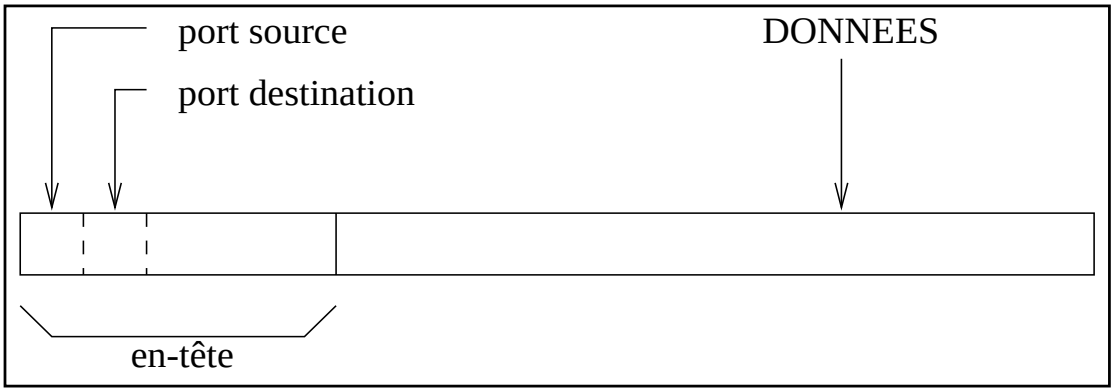

Figure 2.11 *Le paquet TCP*

nom du service	port (protocole)	fonction
ftp-data	20 (TCP)	échange de fichiers (données)
ftp	21 (TCP)	échange de fichiers (commandes)
telnet	23 (TCP)	connexion distante
smtp	25 (TCP)	messagerie SMTP (envoi de messages)
domain	53 (TCP et UDP)	service DNS
http	80 (TCP)	service Web
pop3	110 (TCP)	messagerie (consultation de messages)
nntp	119 (TCP)	forums

Tableau 2.5 *Les ports*

2.5 L'acheminement des données

Quatre types d'équipements distincts permettent d'acheminer les données :

- les répéteurs,
- les ponts (*bridges*),
- les routeurs,
- les passerelles (*gateways*).

Chacun agit au niveau d'une couche distincte, et possède au moins deux interfaces, connectées à des réseaux physiques distincts.

2.5.1 Les répéteurs

Les répéteurs possèdent exactement deux interfaces, et agissent au niveau de la couche physique. Ils amplifient le signal capté sur une interface, et le retransmettent sur l'autre. Les interfaces des répéteurs ne possèdent donc pas d'adresse physique, et *a fortiori* pas d'adresse IP. La figure 2.12 page suivante présente le principe du répéteur.

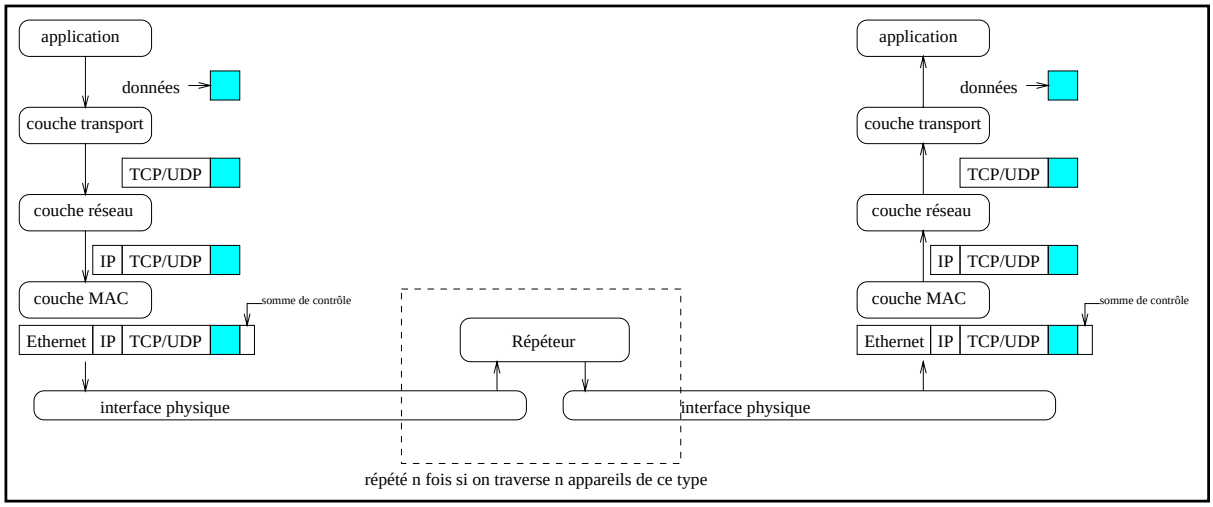

Figure 2.12 *Principe du répéteur*

2.5.2 Les ponts

Les ponts possèdent au moins deux interfaces, et agissent au niveau de la couche liaison de données. Leur rôle est donc de récupérer les trames qui arrivent sur une de leurs interfaces, et de les retransmettre sans modification sur une ou plusieurs de leurs autres interfaces. La figure 2.13 présente le principe du pont.

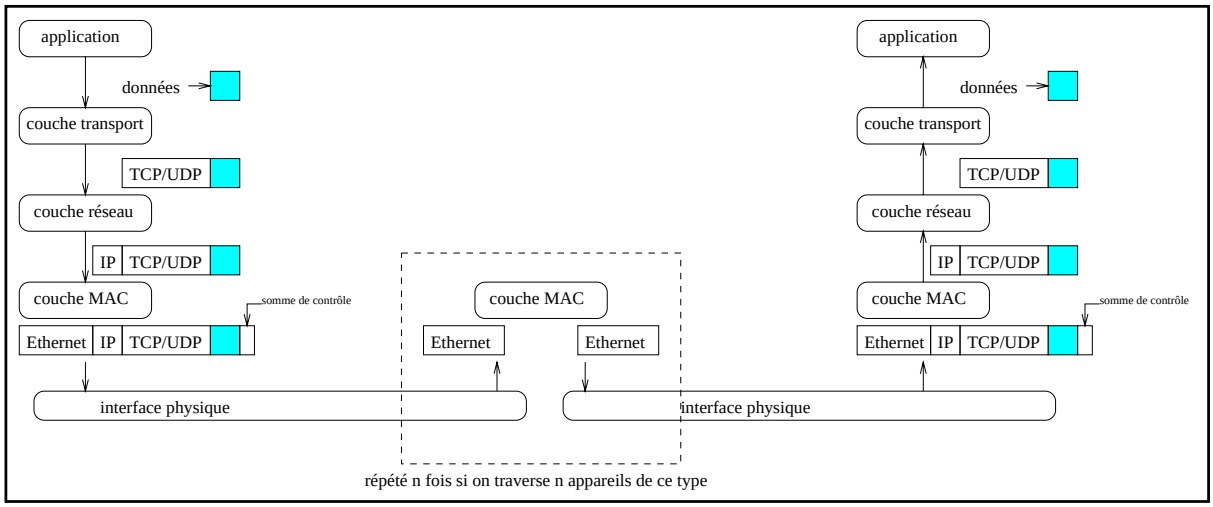

Figure 2.13 *Principe du pont*

Certains ponts savent déterminer, pour chaque nœud destinataire, sur quelle interface renvoyer les trames. Cela évite de les renvoyer sur plusieurs interfaces et produit ainsi une économie de bande passante importante. Ceci est permis par l'examen des paquets qui les traversent ou par l'utilisation de protocoles de communication entre ponts. Ce type d'équipement s'appelle un pont filtrant.

2.5.3 Les routeurs

Les routeurs possèdent au moins deux interfaces, et agissent au niveau de la couche réseau. Leur rôle est donc de récupérer les trames qui arrivent sur une de leurs interfaces, d'en extraire les datagrammes IP, et de les renvoyer sur une autre interface, encapsulés dans de nouvelles trames MAC. Pour déterminer l'interface sur laquelle un paquet doit être réémis, les routeurs consultent leurs tables de routage IP. La figure 2.14 présente le principe du routeur.

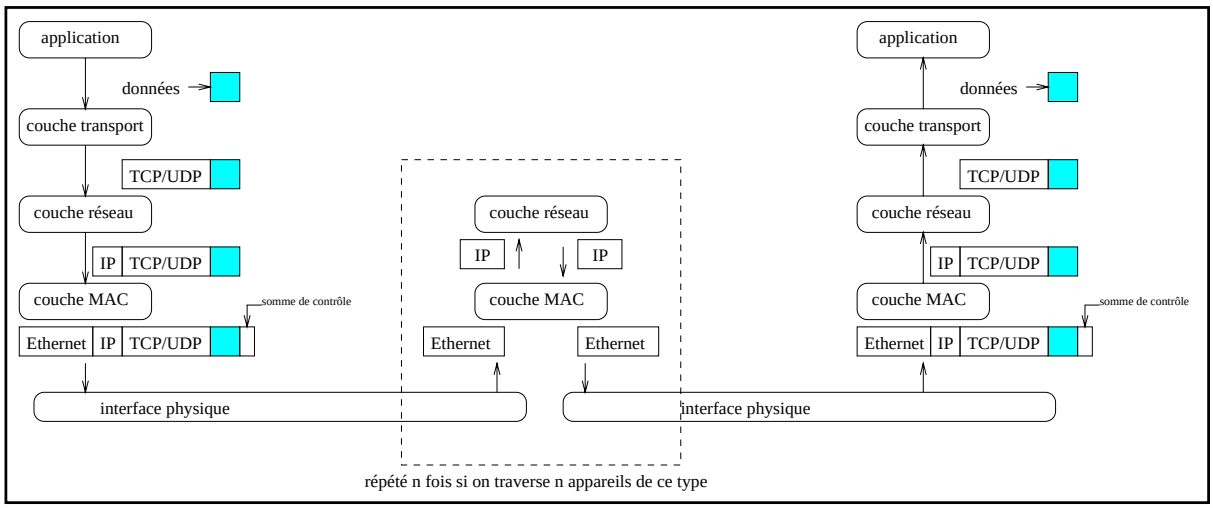

Figure 2.14 *Principe du routeur*

2.5.4 Les passerelles

Les passerelles agissent au niveau de la couche application. Leur rôle est d'intercepter les informations au plus haut niveau, et de les retransmettre vers une autre machine. La figure 2.15 page suivante présente le principe de la passerelle.

2.6 Les protocoles de routage

2.6.1 L'algorithme de routage IP

Le travail d'un routeur consiste, pour chaque datagramme IP récupéré sur une interface, à le renvoyer, par une de ses autres interfaces, vers un routeur plus proche de la destination.

Deux cas se présentent :

– Si l'adresse IP destination contenue dans le datagramme appartient à un même réseau que l'adresse d'une de ses interfaces, le routeur se contente de délivrer le datagramme sur l'interface correspondante, à moins qu'une route très spécifique, pour cette adresse

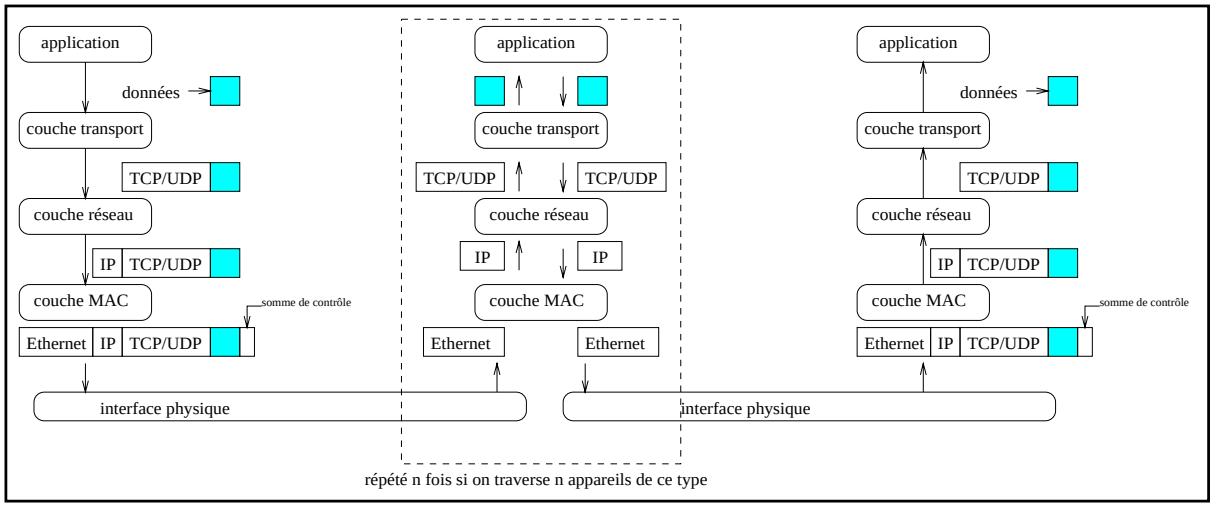

Figure 2.15 *Principe de la passerelle*

IP, soit présente dans sa table de routage. Éventuellement une requête ARP sera émise afin de connaître l'adresse MAC correspondant à l'adresse destination en question.

– Si l'adresse IP destination contenue dans le datagramme n'appartient à aucun réseau directement connecté, le routeur va essayer de déterminer l'interface sur laquelle il doit émettre le datagramme. Pour cela, il vient confronter l'adresse IP destination avec sa table de routage. Celle-ci indique, pour chaque adresse de réseau ou de sous-réseau, l'adresse IP d'un routeur directement connecté, permettant de s'en rapprocher, et une route dite « route par défaut » qui est utilisée en cas de dernier ressort. Notre routeur n'a donc plus qu'à rechercher celle de ses interfaces qui possède une adresse réseau identique à l'adresse réseau du prochain routeur extraite de la table de routage. Il détermine ainsi l'interface physique connectée au prochain routeur. Il se contente pour finir d'envoyer le datagramme à ce dernier dont il connaît maintenant suffisamment d'informations pour être en mesure de le joindre. Il faut noter que notre routeur émettra éventuellement une requête ARP, car la table de routage lui fournit l'adresse IP du prochaine routeur, mais pas son adresse MAC. Un cache sur les résultats des requêtes ARP permet d'en réduire le nombre.

Résoudre le problème du routage consiste à remplir correctement la table de routage. Deux niveaux hiérarchiques sont mis en place sur l'Internet : un premier niveau de routage est mis en place à l'intérieur de chaque fournisseur, et un second niveau est mis en place entre les différents fournisseurs.

2.6.2 Le routage interne à un réseau local

On peut remplir la table de routage de deux manières : par des routes statiques, et par des routes dynamiques.

Le routage statique consiste à recenser manuellement les différents réseaux et sous-réseaux utilisés sur le réseau local, et à configurer manuellement chaque équipement, en lui indiquant, pour chaque réseau, le prochain routeur à contacter pour acheminer les datagrammes.

Le routage dynamique consiste à utiliser des protocoles sur les différents routeurs du réseau local, afin de leur permettre de se configurer automatiquement. Lorsqu'on possède plus de trois ou quatre sous-réseaux, ce type de routage est fortement conseillé.

Les protocoles de ce type sont appelés protocoles de routage interne, ou IGP (Internal Gateway Protocols). Il en existe deux grandes classes : les protocoles à vecteur-distance, tels que RIP ou IGRP, et les protocoles à états des liens, tels qu'OSPF.

Les protocoles à vecteur-distance

Un routeur mettant en jeu un protocole à vecteur-distance se contente d'avertir les autres routeurs directement connectés sur ses interfaces, de la totalité des réseaux qu'il sait atteindre, ainsi que de la distance à laquelle ils se trouvent. Chacun des routeurs qui reçoit cette information la stocke dans sa table de routage, et ajoute aux distances fournies celle qui le sépare du routeur qui lui fait l'annonce. Ainsi, chaque routeur commence par annoncer les réseaux qui lui sont directement connectés, et, de proche en proche, chaque route se retrouve annoncée sur tous les routeurs.

Les trois principaux protocoles à vecteur-distance sont RIP, IGRP et EIGRP.

RIP a été développé par Xerox, et existe sur de très nombreux équipements. La métrique utilisée par RIP est le nombre de routeurs à traverser pour atteindre la cible. De plus, afin de ne pas surcharger la mémoire du système, RIP ne maintient, pour une destination donnée, que la meilleure route, c'est-à-dire celle pour laquelle le nombre de sauts est le plus faible.

RIP impose un certain nombre de limitations :

- par mesure de stabilité, RIP rend inaccessibles les routes situées à plus de 15 sauts. Cela interdit donc son utilisation dans le cadre de réseaux de grande envergure ;
- RIP impose que le masque de sous-réseaux soit unique pour un même réseau ;
- RIP ne permet pas d'utiliser le premier sous-réseau d'un réseau donné ;
- la métrique de RIP est inadaptée à certains types de réseaux, car elle ne prend pas en compte toutes les composantes d'une liaison : deux liaisons de débits différents comptent par exemple toutes les deux pour un saut dans les calculs de plus court chemin, alors qu'il est souvent plus intéressant d'emprunter la liaison de plus haut débit ou de faire du partage de charge sur les deux liaisons.

IGRP (Interior Gateway Routing Protocol) est un protocole propriétaire développé par Cisco Systems, plus robuste que RIP et possédant moins de limitations. EIGRP (Extended Interior Gateway Routing Protocol) en est une version évoluée.

Voici les principaux avantages d'EIGRP :

- il n'est pas limité, comme RIP, à 15 sauts maximum ;
- il a été mis en place et fonctionne parfaitement sur des réseaux de grande envergure ;
- EIGRP est un protocole à vecteur-distance, il utilise donc un ensemble de métriques pour déterminer les routes les plus courtes. Pour cela, il se base sur les délais, le débit, la fiabilité et la charge du réseau.

Les protocoles à état des liens

Les protocoles à état des liens, comme OSPF (Open Shortest Path First), sont chargés d'annoncer l'état des liens physiques, ce qui permet à tous les routeurs de connaître la topologie complète du réseau, et donc de calculer le plus court chemin selon des critères divers.

OSPF a été développé par l'IETF (Internet Engineering Task Force) pour succéder à RIP. Il divise le réseau en zones, chacune étant constituée d'un ensemble de réseaux de machines connexes. Ces différentes zones sont reliées par une épine dorsale (*backbone*) qui comprend le réseau des routeurs interconnectant les différentes zones. Cette épine dorsale est chargée de distribuer les informations de routage entre les différentes zones. La topologie de chaque zone est ainsi invisible pour les autres zones.

2.6.3 Le routage interne à un fournisseur Internet

Un fournisseur Internet met en place une interconnexion entre les réseaux de ses clients.

Avant de vouloir échanger des datagrammes avec les autres fournisseurs, il faut déjà pouvoir échanger des datagrammes entre les clients du même fournisseur. Il existe pour cela principalement deux politiques de routage chez les fournisseurs Internet :

- Les clients choisissent un protocole de passerelle interne pour leur site et mettent en place une route par défaut vers le fournisseur ; ce dernier choisit un protocole de routage pour son réseau qui interconnecte ses clients, et configure ses routeurs raccordant ses clients pour qu'ils s'annoncent entre eux les routes de ces derniers. Il s'agit donc de la mise en place d'un protocole de routage interne au fournisseur, qui n'interopère pas avec ses clients. Quand un équipement d'un client émet un datagramme à destination d'un autre client, ou à destination d'un réseau chez un autre fournisseur, c'est la route par défaut qui entre en jeu pour diriger les données vers le réseau du fournisseur. La figure 2.16 page ci-contre décrit une telle politique de routage.
- Les clients choisissent un protocole de passerelle interne pour leur site ; le fournisseur choisit un protocole de passerelle interne pour son réseau d'interconnexion, et demande à ses clients de configurer leurs routeurs pour interopérer avec les siens. Ainsi, c'est le fournisseur qui annonce, à chacun de ses clients, la route par défaut ainsi que la liste des routes des autres clients. Quand un équipement d'un client émet un datagramme

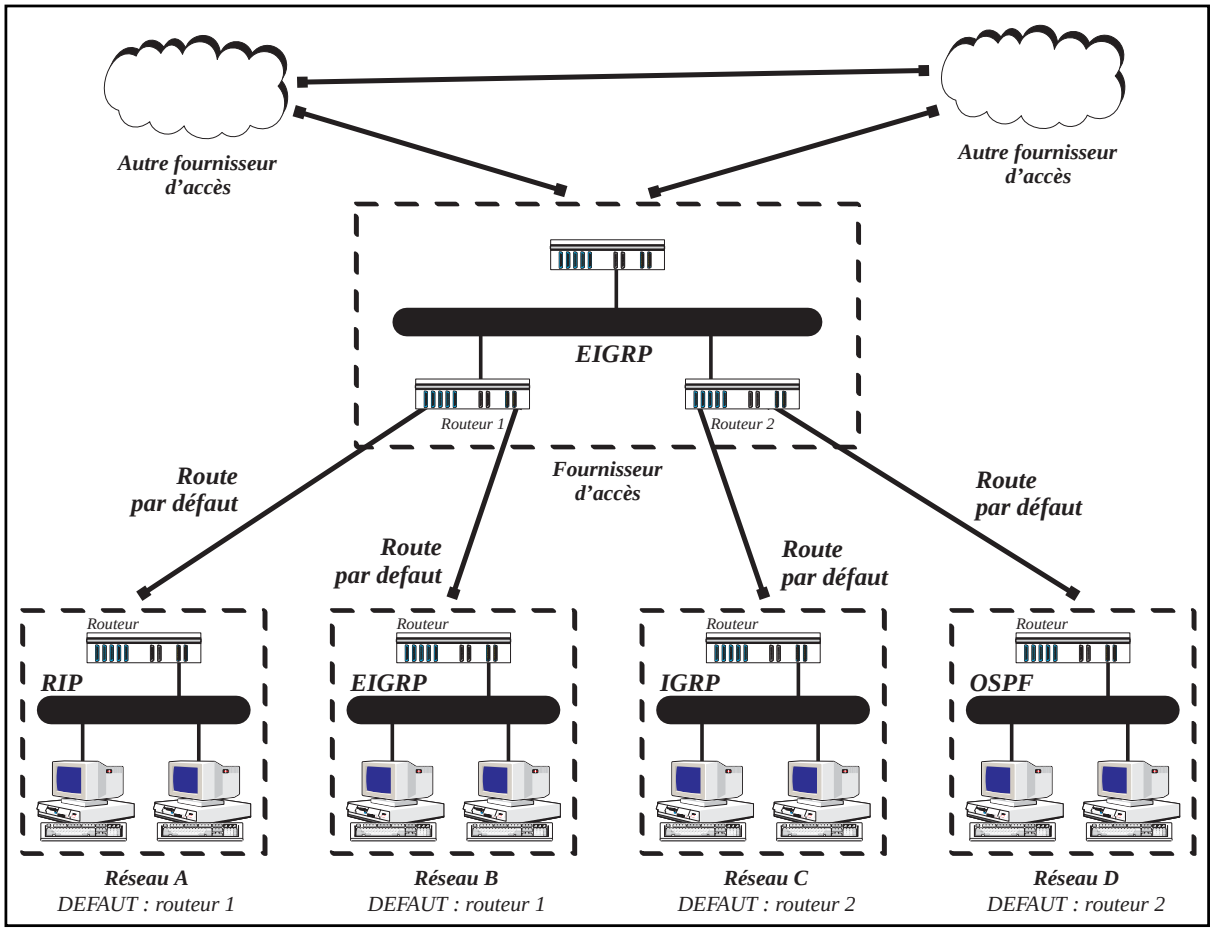

Figure 2.16 *Première politique de routage*

à destination d'un réseau chez un autre fournisseur, c'est la route par défaut qui entre en jeu pour diriger les données vers le réseau du fournisseur ; par contre, lorsque ce datagramme a pour destination un réseau d'un autre client, la route correspondante doit être dans la table de routage du routeur initial. La figure 2.17 page suivante décrit une telle politique de routage, qui est assez peu utilisée par les fournisseurs Internet, car elle impose un travail supplémentaire de configuration chez leurs clients.

2.6.4 Le routage entre fournisseurs

Les protocoles qui permettent aux fournisseurs d'échanger les informations d'accessibilité des réseaux de leurs clients s'appellent des protocoles de passerelle externe. EGP a été le premier protocole de passerelle externe à être utilisé sur l'Internet, mais de nos jours, les fournisseurs utilisent pour la grande majorité le protocole BGP dans sa version 4.

Pour pouvoir mettre en place les protocoles de passerelle externe, il a fallu définir le concept de Système Autonome ou AS (Autonomous System). Chaque fournisseur Internet se voit attribuer, par l'InterNIC, un numéro unique, appelé numéro de système autonome. À chaque

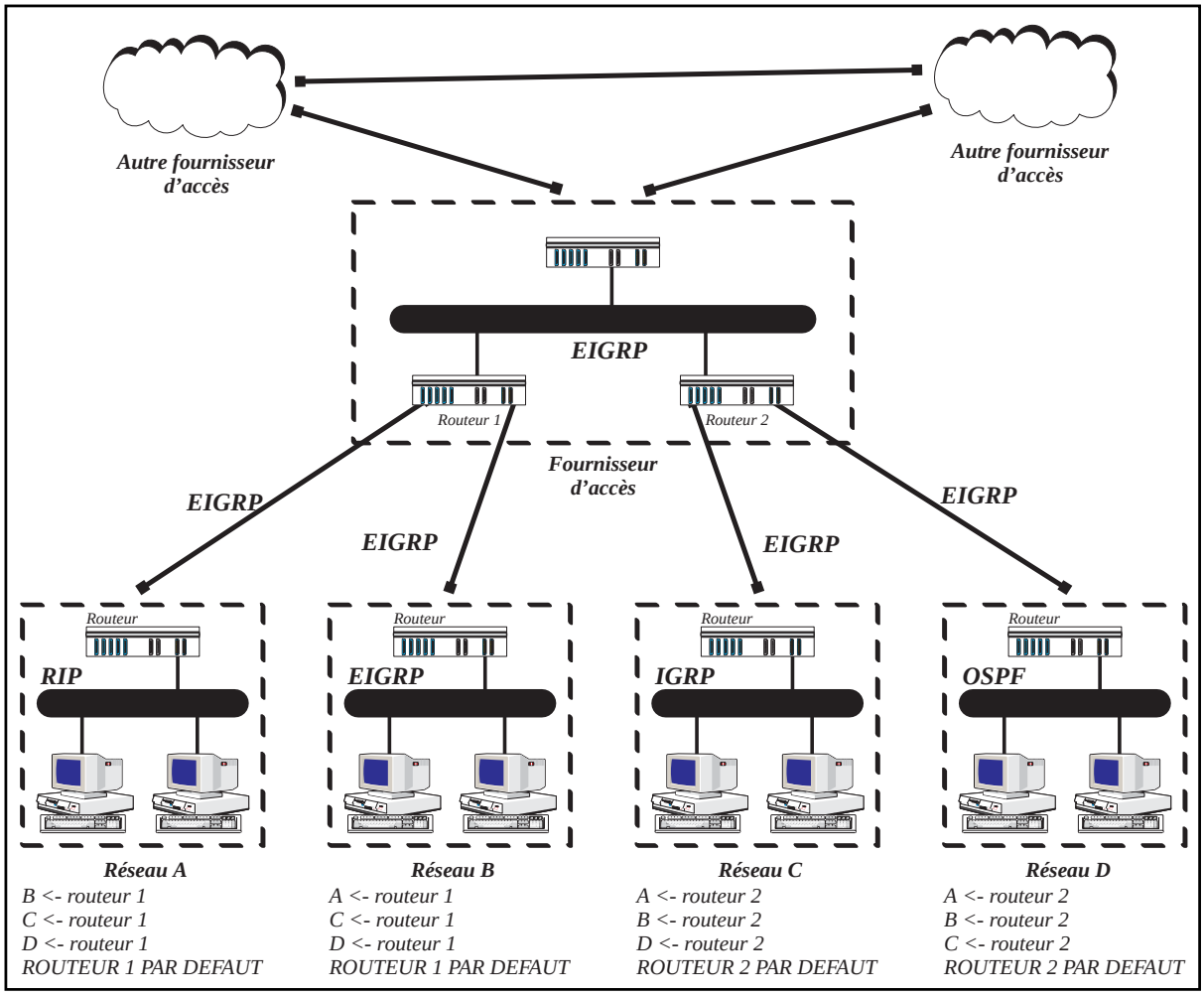

Figure 2.17 *Deuxième politique de routage*

route qui transite sur un protocole de passerelle externe, on associe le numéro d'AS d'origine.

La notation CIDR

Avec BGP-4, il a fallu définir un autre concept : le routage sans classe (CIDR, Classless InterDomain Routing). Nous avons vu jusqu'à présent qu'à chaque réseau, on peut associer un masque de sous-réseaux afin de distinguer des entités plus petites : les sous-réseaux. CIDR propose l'inverse, c'est-à-dire la possibilité de faire des agrégats de réseaux, en faisant disparaître du coup la notion de classe. On va ainsi définir une route CIDR comme un agrégat de réseaux, caractérisé par une adresse CIDR munie d'un masque. L'ensemble des réseaux englobés dans ce masque appartiennent ainsi à cet agrégat. De plus, comme un masque de réseau, ce masque est caractérisé uniquement par le nombre de bits à 1 qui le composent. On va donc prendre la convention de définir ce masque par ce nombre plutôt que de l'écrire en entier. Quelques exemples d'agrégats CIDR sont fournis dans le tableau 2.6 page ci-contre.

Le protocole BGP-4 est décomposé en deux protocoles : iBGP (Internal Border Gateway Protocol) et eBGP (External Border Gateway Protocol). Souvent, plusieurs machines d'un même

notation CIDR	masque correspondant	réseau(x) couvert(s)
10.0.0.0/8	255.0.0.0	10.0.0.0
10.0.0.0/7	254.0.0.0	10.0.0.0, 11.0.0.0
8.0.0.0/6	252.0.0.0	8.0.0.0, 9.0.0.0, 10.0.0.0, 11.0.0.0
172.22.0.0/16	255.255.0.0	172.22.0.0
172.22.0.0/15	255.254.0.0	172.22.0.0, 192.23.0.0
172.20.0.0/14	255.252.0.0	172.20.0.0, 172.21.0.0, 172.22.0.0, 172.23.0.0
192.168.88.0/24	255.255.255.0	192.168.88.0
192.168.88.0/23	255.255.254.0	192.168.88.0, 192.168.89.0
192.168.88.0/22	255.255.252.0	192.168.88.0, 192.168.89.0, 192.168.90.0, 192.168.91.0
192.168.88.0/21	255.255.248.0	192.168.88.0, 192.168.88.89, 192.168.90.0, 192.168.91.0, 192.168.92.0, 192.168.93.0, 192.168.94.0, 192.168.95.0

Tableau 2.6 *Agrégats*

fournisseur gèrent BGP. Ainsi, elles ont besoin de synchroniser leurs tables, c'est le rôle d'iBGP. Les machines qui possèdent les liaisons avec d'autres fournisseurs utilisent iBGP entre elles et eBGP avec les fournisseurs auxquels elles sont directement connectées.

Le principe de BGP est d'établir une table de routage contenant une liste d'agrégats au format CIDR, et pour chacun d'eux le chemin d'AS à traverser afin de l'atteindre, appelé AS-Path.

La métrique BGP permettant de déterminer le chemin le plus court entre deux routes à destination d'un même réseau est la longueur de l'AS-Path, c'est-à-dire le nombre de fournisseurs à traverser pour atteindre la destination, et non pas le nombre de passerelles séparant de la destination, comme c'est le cas dans beaucoup de protocoles de passerelle interne.

2.7 Techniques de configuration

Nous allons maintenant mettre en pratique les notions que nous avons étudiées précédemment afin de configurer les cartes d'interface d'équipements divers et de mettre en place un protocole de routage simple sur un site pris pour exemple. Nous allons pour cela reprendre le plan d'adressage du site présenté sur la figure 2.6 page 39. Rappelons que ce site possède un réseau de classe C, `192.168.22.0`, découpé en trois sous-réseaux avec un masque de valeur `255.255.255.224`. Le lecteur est prié de se référer à la section 2.2.2 page 37 pour une description complète de ce réseau local.

2.7.1 Configuration des adresses IP

Le réseau dispose de Macintosh, de PC et de divers routeurs. Nous allons étudier la configuration de ces différents équipements afin de leur indiquer une adresse IP, le masque de sous-réseaux ainsi que leur route par défaut. Il est nécessaire d'indiquer une route par défaut pour les Macintosh et les PC car, le plus souvent, les logiciels réseau qui les équipent sont incapables d'apprendre la route par défaut par l'intermédiaire d'un protocole de routage.

Macintosh

Sur Macintosh, on trouve différents gestionnaires de pile TCP/IP. Le plus courant s'appelle MacTCP. Une pile plus récente, OpenTransport, est disponible à partir du système 7.5.2 ou 7.5.3 selon les modèles de Macintosh, mais MacTCP reste un bon gestionnaire réseau, suffisamment léger lorsqu'on dispose d'un Macintosh moyen de gamme.

Pour configurer le Macintosh d'adresse `192.168.22.65` de la figure 2.6 page 39, il faut ouvrir le tableau de bord MacTCP et rentrer les différentes informations afin d'obtenir l'écran représenté sur la figure 2.18 page ci-contre.

La route par défaut correspond à une des adresses IP du routeur connectant ce brin Ethernet au reste du réseau. Plus précisément, c'est l'adresse IP de la carte Ethernet de ce routeur située sur ce brin : `192.168.22.68`.

Windows

Microsoft ne fournit pas de pile TCP/IP avec Windows 3.1. Par contre, avec Windows 95 et Windows NT, on trouve des piles TCP/IP livrées avec le système. Mais le nombre d'outils fournis avec la pile reste réduit à sa plus simple expression. C'est pourquoi l'utilisateur d'un système d'exploitation de Microsoft est souvent amené à faire l'acquisition d'une pile TCP/IP commerciale. Elles sont pour la plupart munies de multiples outils permettant d'accéder par exemple au Web, aux forums ou aux services multicast audio et vidéo.

Nous allons étudier la configuration de Chameleon de Netmanage, une pile TCP/IP disponible pour Windows 3.1, Windows 95 et Windows NT, munie de toute une gamme d'outils spécialement conçus pour l'accès aux services disponibles sur l'Internet.

Configurons donc le PC situé sur le sous-réseau `192.168.22.96`. Il faut procéder pour cela en 5 étapes :

1. on commence par créer une interface Ethernet en chargeant l'application Custom et en choisissant le menu ***[Interface/Add]*** (figure 2.19 page suivante) ;
2. on entre alors les paramètres matériels tels que le numéro d'interruption de la carte ou son adresse IO, depuis le menu ***[Setup/Hardware]*** (figure 2.20 page 58) ;

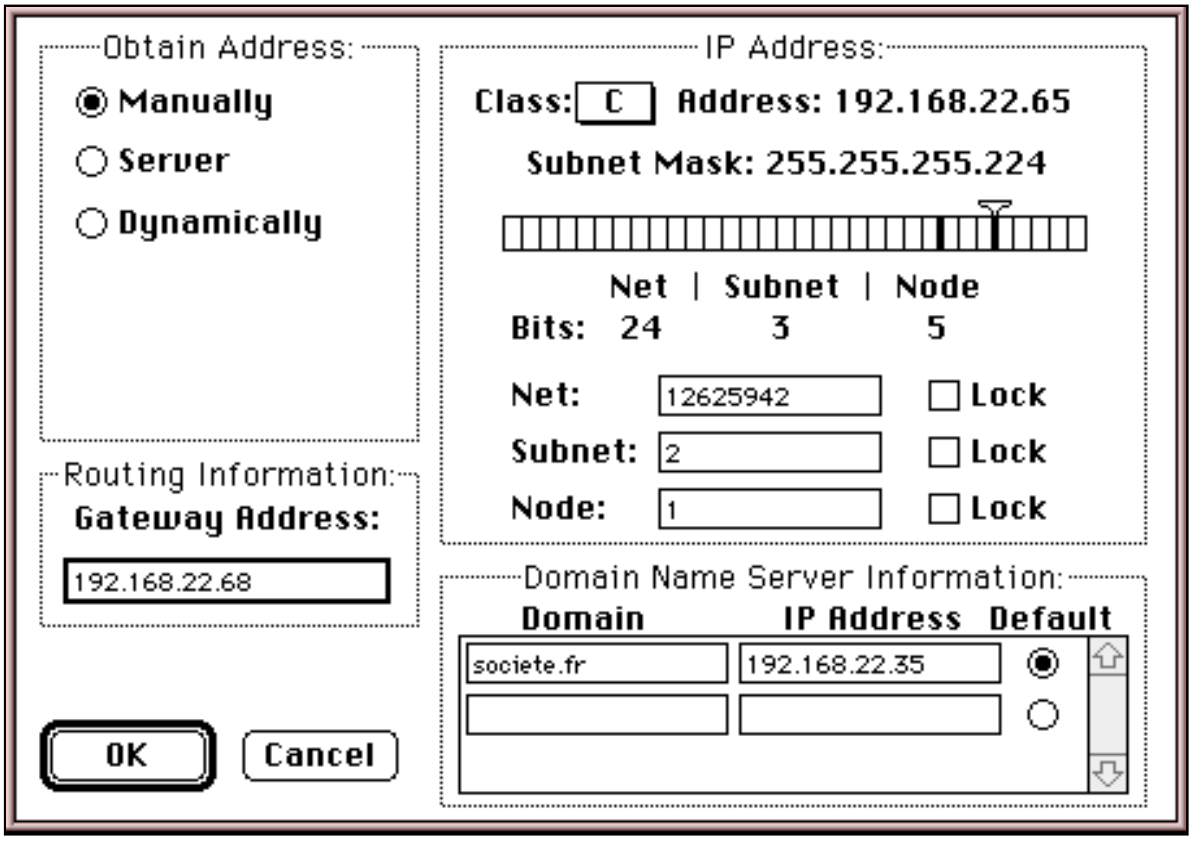

Figure 2.18 *Configuration TCP/IP sur Macintosh*

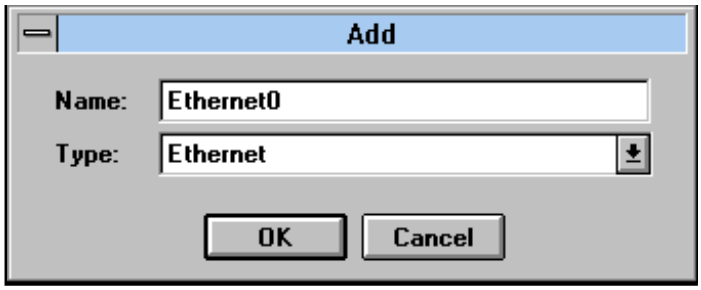

Figure 2.19 *Création d'une interface Ethernet*

3. avec l'onglet ***[Setup/Configuration/IP Configuration]***, on saisit l'adresse IP ainsi que le masque de sous-réseaux (figure 2.21 page suivante) ;

4. l'onglet ***[Setup/Configuration/Gateway]*** permet de positionner la route par défaut (figure 2.22 page 59) ;

5. on peut maintenant vérifier, en revenant au premier écran de l'application Custom, que l'interface est correctement configurée (figure 2.23 page 59).

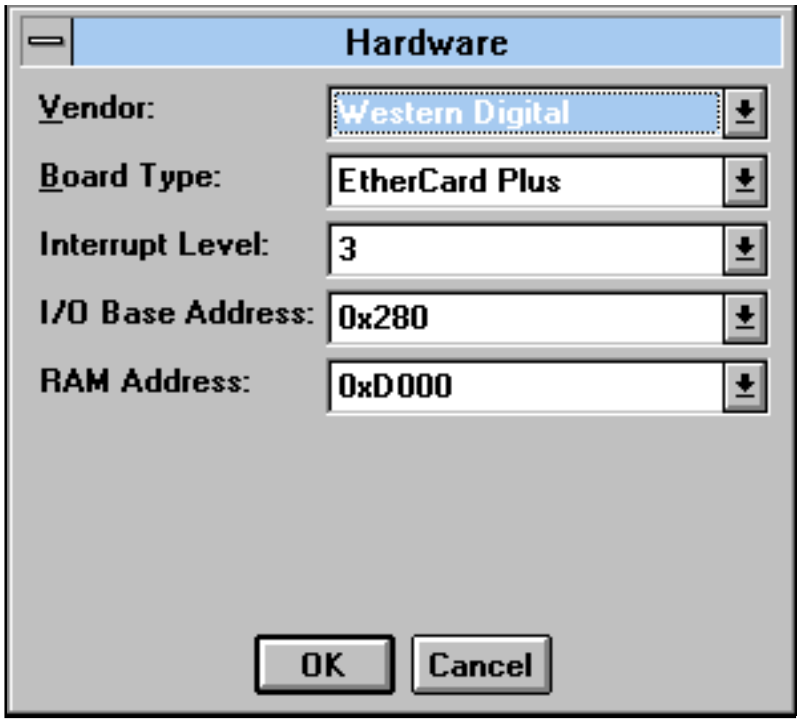

Figure 2.20 *Configuration matérielle d'une interface Ethernet*

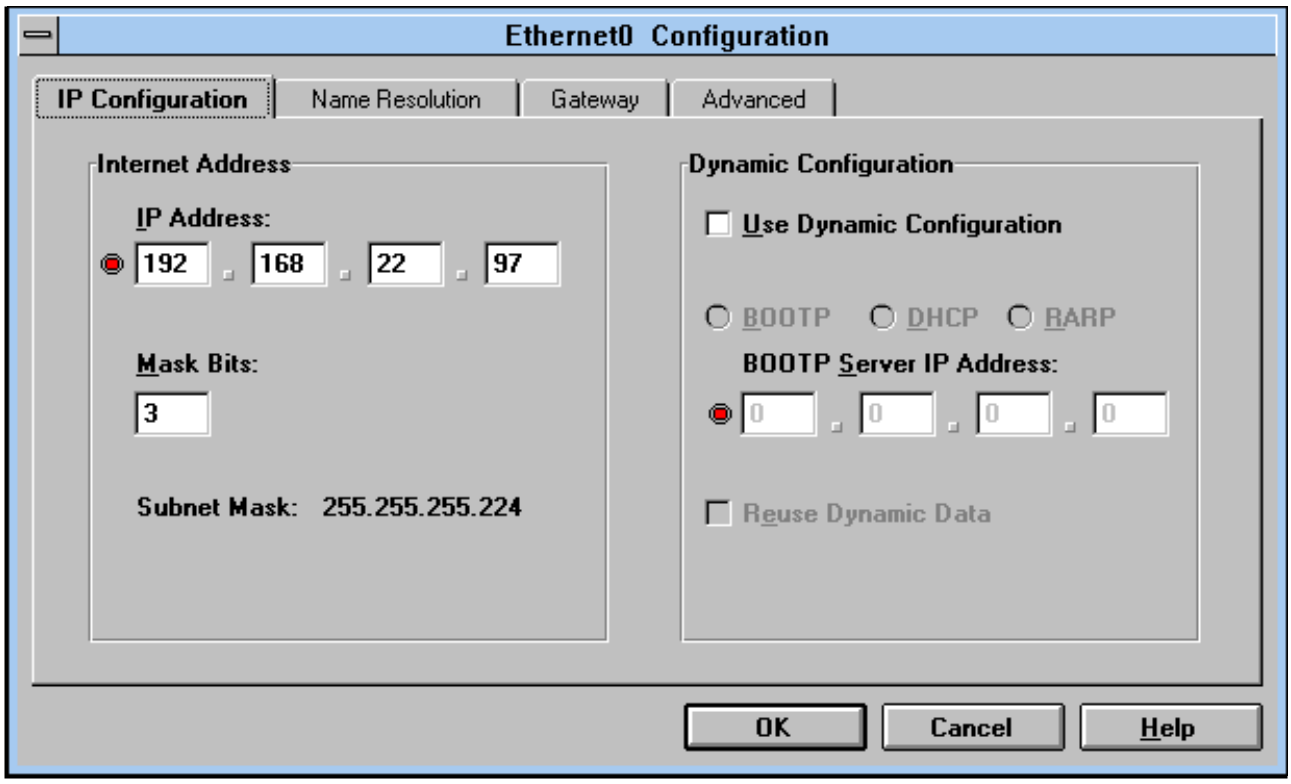

Figure 2.21 *Configuration IP d'une interface Ethernet*

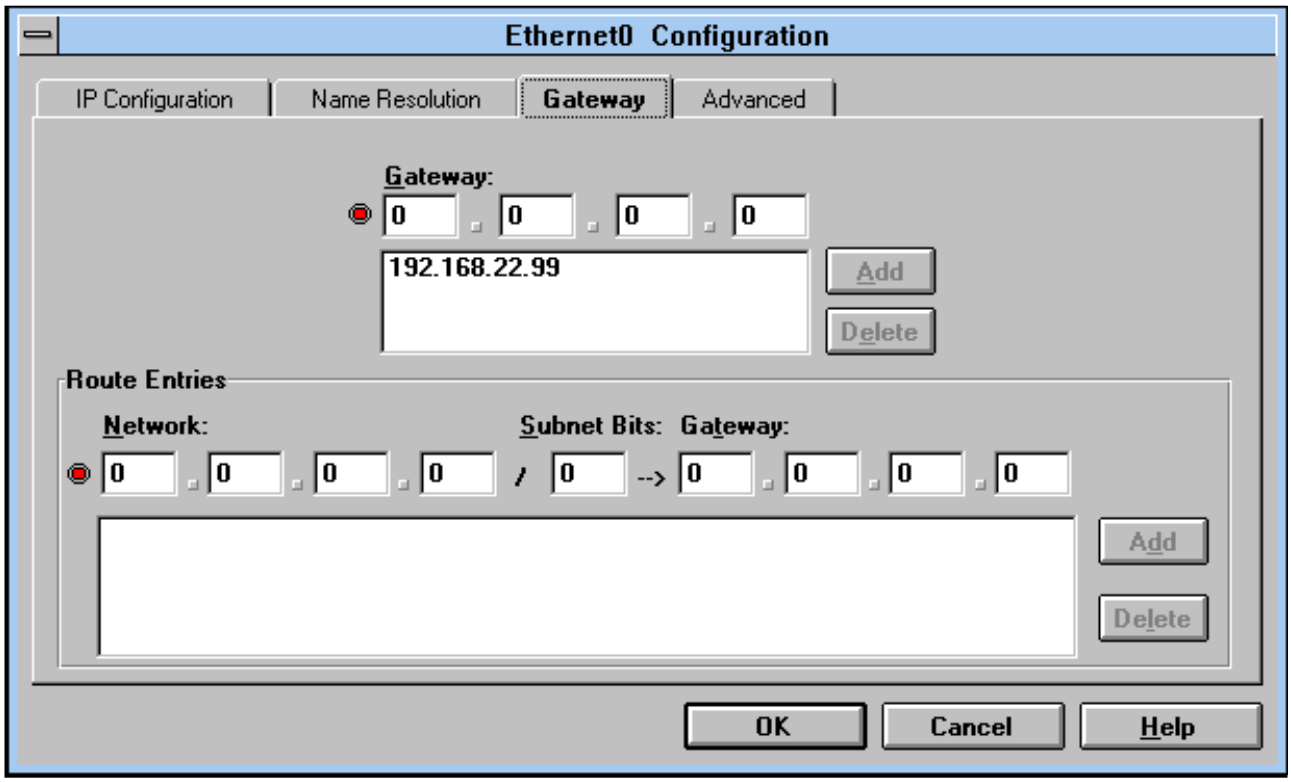

Figure 2.22 *Configuration du routage*

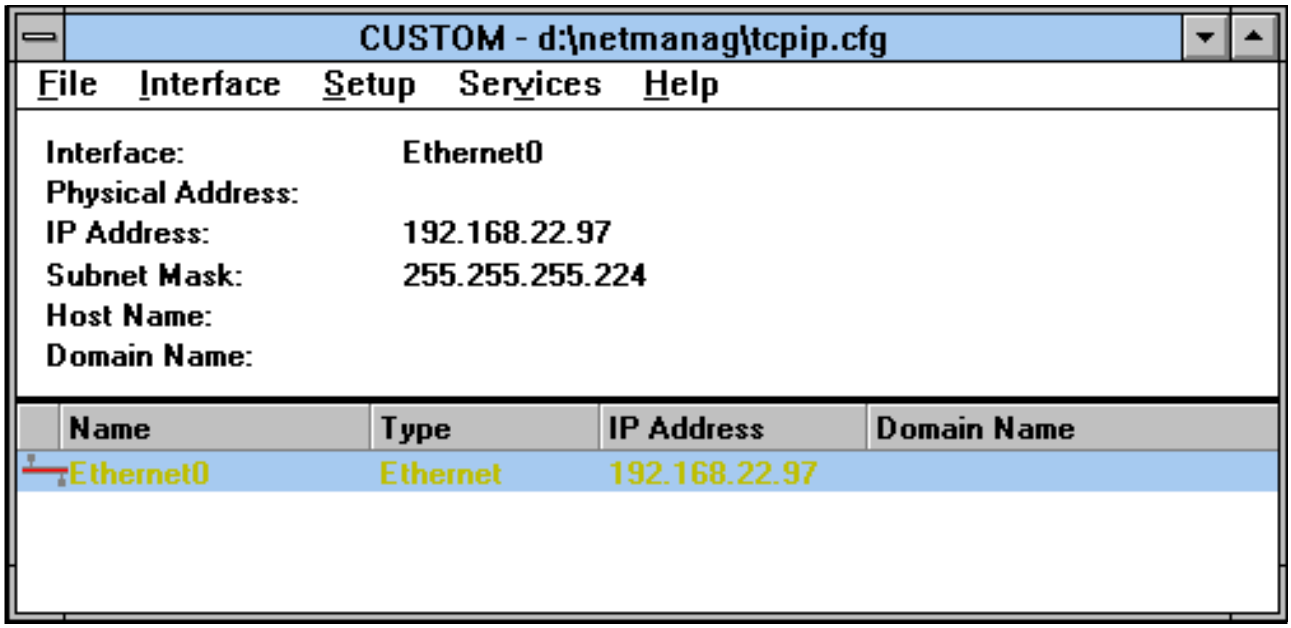

Figure 2.23 *Application Custom : configuration d'une interface*

Unix

Attachons-nous maintenant à configurer les serveurs Unix situés sur le sous-réseau d'adresse `192.168.22.32`. Supposons par exemple qu'il s'agisse d'une station de travail Sparc sous Solaris 2.5. Notons que la configuration avec un autre système Unix serait similaire.

Nous allons donc configurer la station de travail dont l'adresse IP est `192.168.22.35`. Nous allons la nommer `maelle` et la placer dans le domaine `fenetre.fr`.

Le fichier `/etc/defaultdomain` contient le domaine.

Le fichier `/etc/hostname.`{`nom de l'interface Ethernet`} (dans notre cas il s'agit du fichier `/etc/hostname.le0`) contient le nom de la machine.

la commande `ifconfig -a` permet de connaître le nom des interfaces disponibles.

Nous mettons donc en place ces deux fichiers comme suit :

```
% su -
Password:
# echo fenetre.fr > /etc/defaultdomain
# echo maelle > /etc/hostname.le0
# ^D
%
```

Le fichier `/etc/hosts` doit contenir l'adresse IP de l'interface `le0`. Il contient donc :

```
127.0.0.1       localhost
192.168.22.35   maelle maelle.fenetre.fr
```

Le masque de sous-réseaux est indiqué dans le fichier `/etc/netmasks` qui doit ainsi contenir la ligne suivante :

```
192.168.22.0    255.255.255.224
```

Il suffit maintenant de redémarrer la machine et les modifications vont être prises en compte :

```
% su -
Password:
# touch /reconfigure
# reboot
```

On peut le constater avec la commande `ifconfig` :

```
% ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 192.168.22.35 netmask ffffffe0 broadcast 192.168.22.63
```

On notera qu'il n'a pas fallu mettre en place de route par défaut, c'est le démon de routage qui

s'en chargera. Si on n'utilise pas de démon de routage, le fichier `/etc/defaultrouter` doit contenir le nom du routeur par défaut (ici le nom de domaine correspondant à l'adresse IP `192.168.22.33`) afin que la route par défaut soit mise en place au démarrage.

Routeur

Les routeurs comme celui qui raccorde les sous-réseaux d'adresses `192.168.22.64` et `192.168.22.32` sont souvent des machines Unix ou des matériels dédiés.

Dans le cadre d'une machine Unix, pour configurer les différentes interfaces du routeur, il faut connaître leurs noms (commande `ifconfig -a`), par exemple `le0` et `le1`, et remplir comme on l'a fait dans la section précédente les fichiers `/etc/hostname.le0`, `/etc/hostname.le1`, `/etc/hosts` et `/etc/netmasks`. Il faut enfin redémarrer la machine.

Dans le cadre d'un routeur spécialisé, il faut se référer à la documentation du constructeur.

2.7.2 Configuration du routage

Nous allons nous attacher à configurer le routage, c'est-à-dire à remplir de manière automatique la table de routage de chacun de nos équipements, stations de travail et routeurs. Ce n'est pas utile pour les PC et Macintosh car on leur a indiqué une route par défaut vers un routeur directement connecté.

Unix

Sur les machines Unix, qu'elles soient simples serveurs ou qu'elles fassent aussi office de routeurs, il suffit d'activer le démon de routage. Les deux démons de routage sous Unix sont `routed` et `gated`. `routed`, livré avec la plupart des systèmes Unix, autorise uniquement le protocole RIP tandis que `gated`, rarement fourni avec le système Unix, est compatible avec la plupart des protocoles de routage internes et externes pour TCP/IP.

On peut rapatrier `gated` par FTP à l'URL suivant :
`ftp://ftp.ibp.fr/networking/gated`

`routed` est lancé si nécessaire au démarrage, sous Solaris. On peut lui indiquer dans le fichier `/etc/gateways` les différents routeurs avec lesquels il doit échanger des informations. En l'absence de ce fichier, `routed` peut quand même échanger les informations de routage avec les routeurs voisins. Il est activé avec le paramètre `-q` pour capturer les routes sans faire d'annonce. Sur une machine multi-domiciliée faisant office de routeur, il est activé avec le paramètre `-s` afin non seulement de récupérer des informations sur l'accessibilité des réseaux, mais aussi d'annoncer les réseaux correspondant aux différentes interfaces Ethernet de la machine.

Pour utiliser `gated` sur une machine Unix multi-domiciliée, on peut l'activer en utilisant un fichier de configuration minimum (`/etc/gated.conf`), comme suit :

```
% cat /etc/gated.conf
rip yes {
  interface all ripin ripout ;
} ;
% su -
Password:
# gated &
```

Par contre, sur une machine Unix ne faisant pas office de routeur, on utiliserait un fichier de configuration différent, afin de ne pas faire d'annonce :

```
% cat /etc/gated.conf
rip yes {
  interface all ripin noripout ;
} ;
% su -
Password:
# gated &
```

Routeur spécialisé

Pour activer le routage RIP avec un routeur spécialisé, il faut se conformer à la documentation fournie par le constructeur.

Par exemple, en supposant que le routeur qui relie le sous-réseau de serveurs au sous-réseau de PC soit un matériel CISCO, il suffit de le configurer comme suit :

```
CISCO>enable
Password:
CISCO#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
CISCO(config)#router rip
CISCO(config-router)#network 192.168.22.0
CISCO(config-router)#passive-interface serial 0
CISCO(config-router)#^Z
CISCO#write memory
```

La ligne `passive-interface` permet de demander au routeur de ne pas activer RIP sur l'interface indiquée en paramètre. En effet, lors d'un raccordement de type intermittent, par RNIS ou X25 sur Transpac, il ne faut pas activer le protocole de routage sur les interfaces WAN car les paquets engendrés maintiendraient ainsi la connexion active même si aucun autre trafic n'y transitait.

3

Raccordement d'un réseau local à l'Internet

Dans le cadre du raccordement d'un réseau local à l'Internet, l'offre des fournisseurs comporte une ou plusieurs des méthodes que nous allons décrire dans ce chapitre : connexion par modem sur le réseau téléphonique commuté, connexion par RNIS (réseau numérique à intégration de services), location d'une ligne spécialisée, connexion par X25 sur Transpac.

3.1 Généralités

3.1.1 Les quatre offres traditionnelles

Pour se connecter au réseau Internet par un fournisseur, il faut choisir un type de ligne de transmission entre le réseau local et celui du fournisseur.

Les fournisseurs proposent traditionnellement quatre types de raccordements, chacun ayant ses propres caractéristiques.

Connexion par le réseau téléphonique commuté (RTC)

Pour ce type de raccordement, il faut acquérir un modem et le brancher sur un des équipements connectés au réseau local. On va le transformer en routeur IP en choisissant un protocole réseau pour transporter les paquets IP à travers le modem. Pour cela, il faut installer un logiciel correspondant au protocole choisi sur l'équipement. Ce protocole doit être le même

que celui utilisé par le fournisseur. Certains d'entre eux proposent des accès par le protocole SLIP (Serial Link Internet Protocol), mais la plupart proposent à leurs clients d'utiliser PPP (Point to Point Protocol), qui est bien plus robuste.

Connexion par RNIS

Pour ce type de raccordement, il faut acquérir une carte RNIS ou un adaptateur de terminal RNIS qu'il faudra relier à un équipement déjà connecté au réseau local, ou un routeur dédié. Dans la grande majorité des cas, le fournisseur propose d'utiliser le protocole PPP pour transporter les paquets IP entre le réseau local du client et le sien.

Connexion par ligne spécialisée numérique

La raccordement par ligne spécialisée numérique, aussi appelée LS ou ligne louée, est le moyen de connexion le plus agréable pour accéder à l'Internet. Bien qu'une carte synchrone rajoutée à une des machines du réseau local puisse convenir, on fait souvent l'acquisition d'un équipement dédié pour brancher une ligne spécialisée. Cette dernière, tirée par l'opérateur télécom entre le site du client et le fournisseur, fournit un débit garanti et une connexion permanente de l'ensemble des machines, de jour comme de nuit. Le protocole utilisé sur ce type de ligne est souvent, là encore, PPP.

Connexion par X25 sur Transpac

Le raccordement par X25 sur Transpac permet d'utiliser le réseau Transpac pour se relier à un fournisseur. Différentes solutions sont disponibles pour ce type de connexion : équipement dédié ou carte ajoutée à un des équipements déjà connectés au réseau local. On utilise le plus souvent le protocole d'encapsulation IP sur X25. PPP, quant à lui est plus souvent utilisé pour la connexion des particuliers par Transpac, par l'intermédiaire d'un PAD (Packet Assembler Desassembler), dont le principe est décrit à la section 3.7.2 page 132.

3.1.2 Points de Présence

Les fournisseurs Internet en France possèdent pour la plupart un centre d'opérations techniques principal (NOC, Network Operation Center) situé en région parisienne. Or, pour la plupart des services Internet professionnels, le coût de la ligne de transmission auprès de l'opérateur télécom dépend de la distance qui sépare le fournisseur de son client. C'est notamment le cas pour les communications RTC sur modem, RNIS, et les lignes spécialisées ; Transpac propose quant à lui un mode de facturation indépendant de la distance. Ainsi, pour ne pas imputer des coûts de communication trop importants aux entreprises qui se connectent à leurs services depuis la province, les fournisseurs choisissent le plus souvent d'étendre leur

présence sur le territoire par la mise en place de points de présence, appelés POP[1]. Chaque fournisseur en possède souvent plus d'une dizaine, proches des grandes villes de métropole.

Les différents POP d'un même fournisseur sont reliés en étoile par des lignes spécialisées au centre d'opérations principal. Le plus souvent, ces lignes ont un débit de 64 Kbits/s. Chaque POP fournit tout ou partie des services offerts par le centre d'opérations principal (exceptée la connexion X25 sur Transpac qui se fait toujours directement avec le centre d'opérations techniques principal) : accès par modems, RNIS et lignes spécialisées. Cela permet de proposer un numéro d'accès local pour les services intermittents et des LS de courte distance, et donc à bas prix, pour les accès permanents.

Il faut noter qu'un POP relié par une ligne 64 Kbits/s avec le centre parisien accueille bien souvent plusieurs clients en LS 64 Kbits/s. Il s'agit là d'une technique de surréservation[2] très utilisée par les fournisseurs Internet, car bien souvent on n'utilise en moyenne sur l'année que quelques Kbits/s sur une LS 64 Kbits/s.

La figure 3.1 présente un exemple de topologie de fournisseur utilisant des points de présence.

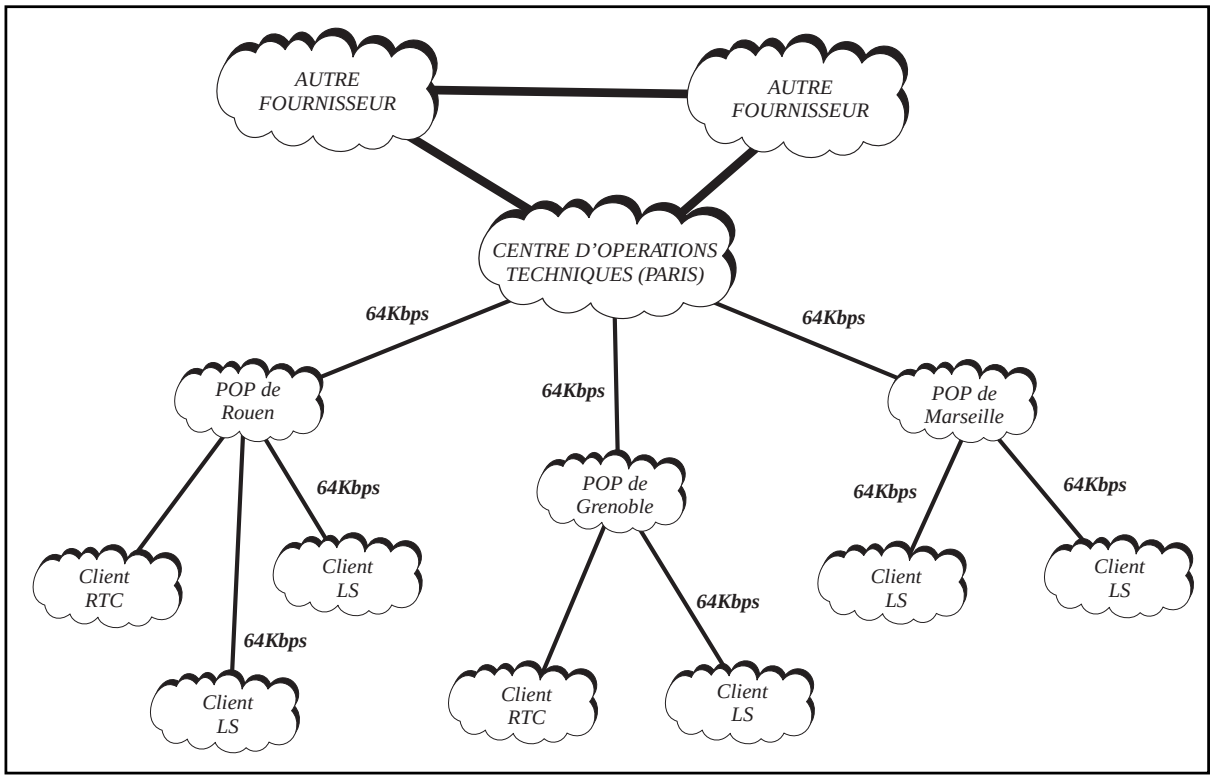

Figure 3.1 *Points de présence d'un fournisseur*

1. POP : Point Of Presence
2. *overbooking*

3.1.3 Équipements

On classe les équipements nécessaires à la connexion distante avec un fournisseur en deux catégories :

- l'ETTD (équipement terminal de transmission de données), ou DTE (data terminal equipment), est connecté au réseau local et assure un certain nombre de services ;
- l'ETCD (équipement terminal de circuit de données), ou DCE (data circuit terminating equipment), est relié à l'ETTD local par une liaison particulière appelée jonction, et à l'ETCD distant par la ligne de transmission.

Ces deux types d'équipements sont présents autant chez le fournisseur que chez ses clients, la figure 3.2 schématise leur utilisation.

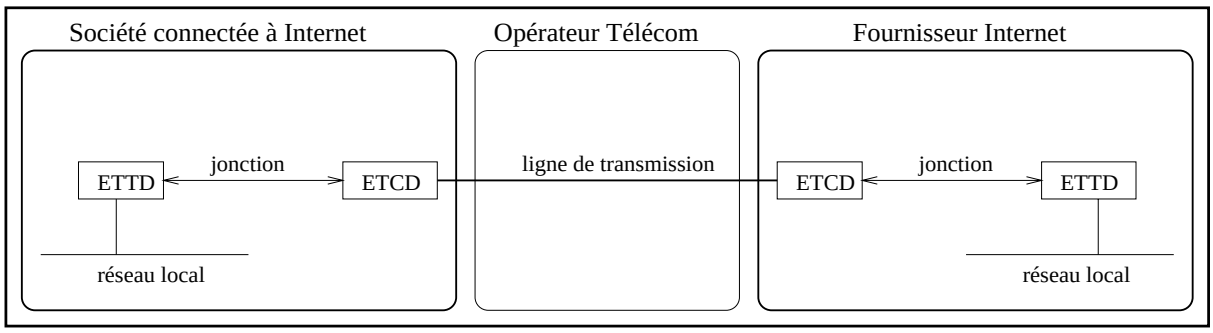

Figure 3.2 *ETTD et ETCD*

Par exemple, dans le cadre d'une connexion Internet par le RTC, c'est un serveur Unix qui pourra jouer le rôle d'ETTD, et un modem branché sur ce dernier jouera le rôle d'ETCD.

3.1.4 Types de jonction et mode de transmission

Pour pouvoir correctement échanger des bits sur une jonction, le récepteur doit être capable d'obtenir un signal d'horloge caractérisant le débit du flux d'émission.

Jonctions synchrones et asynchrones

Les jonctions de type synchrone sont caractérisées par des échanges au niveau bit, l'émetteur étant synchronisé sur une horloge de fréquence fixe. Cette dernière est soit transportée sur un des câbles électriques constituant la liaison, soit régénérée par le récepteur à partir d'une boucle verrouillée en phase, acceptant en entrée le flux de données de la jonction.
Les jonctions de type asynchrone permettent, quant à elles, à l'émetteur, de transmettre des signaux à tout moment. Ils sont ainsi groupés en caractères, délimités par des signaux particuliers appelés START et STOP.
Les liaisons synchrones sont utilisées pour les moyens et hauts débits, par exemple dans le cadre des lignes spécialisées. Les liaisons asynchrones le sont pour les bas débits, par exemple lors d'un raccordement par modem.

Les interfaces physiques sont définies par quatre paramètres : les caractéristiques fonctionnelles, électriques et mécaniques.

Le tableau suivant récapitule cinq des principales interfaces physiques :

Caractéristiques fonctionnelles	**Caractéristiques électriques**	**Caractéristiques mécaniques**	**Utilisation traditionnelle**
V24	V28	DB-9 et DB-25	RTC et LS
V24	V35	DB-34	
X24	V11	DB-9 et DB-15	X25 hauts débits (Transpac)
X21 bis	V28	DB-25	X25 bas débits (Transpac)
I431			RNIS

3.1.5 Protocoles pour le raccordement

Les protocoles utilisés sur la ligne de transmission, dans le cadre du raccordement d'un réseau local à l'Internet, dépendent du type d'accès. Le tableau 3.1 présente, pour chaque type d'accès, les protocoles utilisés le plus souvent.

Type d'accès	**Protocoles courants**	
RTC	**PPP asynchrone**	SLIP
RNIS	**PPP synchrone**	PPP asynchrone
LS	**HDLC**	PPP synchrone
Transpac	**IP/X25**	PAD X3

Tableau 3.1 *Protocoles couramment utilisés dans le cadre d'un raccordement Internet*

3.2 PPP

3.2.1 Principe

Nous l'avons vu précédemment, PPP est un protocole qui s'intercale entre un support de transmission et les datagrammes de la couche IP. Notons qu'il est tout à fait possible d'encapsuler d'autres types de protocoles qu'IP, par exemple Appletalk, Novell IPX, DECnet Phase IV ou bien même ISO CLNP.

Cette section dédiée à PPP va nous permettre de découvrir ce protocole qu'on est souvent amené à mettre en place lors d'une connexion avec un fournisseur Internet, que ce soit pour une liaison intermittente ou permanente.

PPP est composé de trois entités :

1. une méthode pour encapsuler des datagrammes sur des liaisons série ;
2. un protocole de contrôle de liaison évolué : LCP, Link Control Protocol ;
3. une famille de protocoles permettant d'établir et de configurer des couches au niveau réseau [3] : ce sont les NCP (Network Control Protocols).

La figure 3.3 décrit le principe de PPP sur un exemple : le transport d'informations entre un client et un serveur World Wide Web. Les données du protocole HTTP (Hyper Text Transport Protocol : protocole utilisé pour le WWW) sont encapsulées sur le réseau local dans des paquets TCP, eux-mêmes encapsulés dans des datagrammes IP, ces derniers étant enfin encapsulés dans des trames Ethernet. Lorsque ces dernières rencontrent la machine qui dispose d'une passerelle PPP et qui est reliée au modem, les datagrammes IP en sont extraits et encapsulés dans des trames PPP. Le processus inverse est utilisé de l'autre côté de la liaison PPP : les datagrammes IP sont extraits des trames PPP et sont encapsulés dans des trames Ethernet du réseau local du fournisseur.

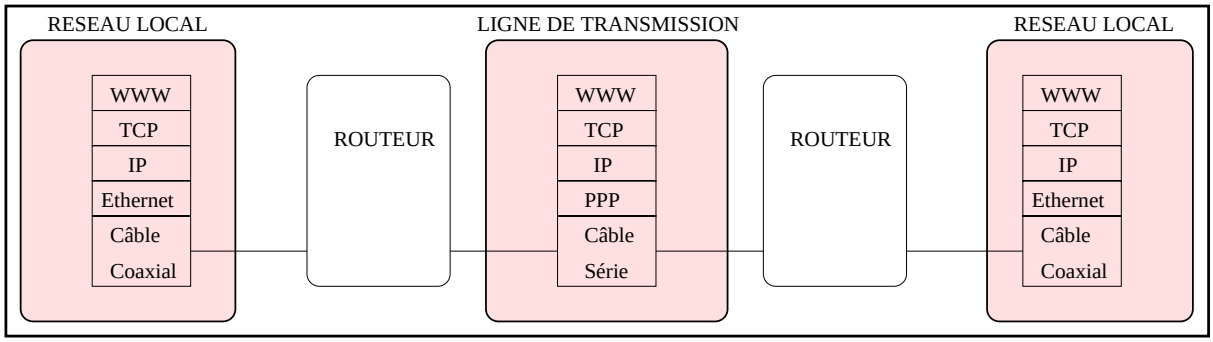

Figure 3.3 *Principe de PPP*

3.2.2 PPP synchrone et asynchrone

PPP peut traverser des jonctions synchrones et asynchrones, c'est-à-dire des liaisons série au niveau bit (liaisons synchrones) et au niveau caractère (liaisons asynchrones). La couche la plus basse du protocole PPP est adaptée au type de jonction. Notamment, l'échappement des caractères de début et de fin de trame n'est pas intégré de la même manière en synchrone et en asynchrone.

Les fournisseurs Internet proposent des connexions PPP asynchrones avec leurs services sur RTC bien qu'il soit tout à fait possible d'utiliser des jonctions synchrones entre les modems et les ETTD, mais il faut dans ce cas que le fournisseur et son client choisissent tous deux ce mode de connexion synchrone, ce qui est rare. Les connexions PPP sont plutôt en mode synchrone avec RNIS, et systématiquement en mode synchrone sur les lignes spécialisées.

3. Niveau 3 du modèle OSI

La plupart des passerelles PPP n'autorisent que les connexions asynchrones. Il faut le vérifier avant de faire l'acquisition d'un tel logiciel, en fonction du type d'accès choisi avec le fournisseur.

3.2.3 Dial-on-Demand

Le Dial-on-Demand n'a de raison d'être que dans le cadre d'une connexion intermittente avec l'Internet (RTC, RNIS). C'est la faculté d'une passerelle PPP à se connecter automatiquement au fournisseur lorsqu'elle capture sur le réseau local du client des paquets à destination de l'Internet. Certaines passerelles PPP n'ont pas cette faculté, et c'est alors une manipulation d'un opérateur qui est nécessaire pour établir la connexion PPP.

De même, la coupure de connexion au bout d'un temps d'inactivité sur la ligne n'est souvent disponible qu'avec les passerelles PPP proposant le Dial-on-Demand. Cette technique est très utilisée avec les connexions RNIS, car contrairement à l'établissement d'une communication sur RTC pour laquelle il faut parfois plus de 30 secondes, la communication sur RNIS peut être activée en moins d'une seconde. Cela permet donc de faire des économies de facture téléphonique en gardant une procédure d'accès à l'Internet transparente pour l'utilisateur.

3.2.4 Redial

Le Redial est la faculté d'une passerelle PPP à se reconnecter automatiquement lorsque la communication est coupée anormalement ou à la demande de la couche LCP. En effet, une connexion PPP peut-être amenée à être coupée par la couche LCP ; celle-ci procède en envoyant des demandes d'echo régulières, et en cas d'absence de réponse, elle peut décider de déconnecter la liaison PPP. Il arrive qu'on résolve certains problèmes de déconnexion intempestive en désactivant cette fonctionnalité de la couche LCP.

3.2.5 Filtres

La notion de filtre intervient à deux niveaux dans une passerelle PPP :

1. La configuration d'une passerelle PPP doit permettre de sélectionner les datagrammes IP qui peuvent la traverser et ceux qui sont rejetés. C'est le nombre de ces critères permettant de configurer la fonction de filtrage, ainsi que leur flexibilité, qui va compter dans le choix d'une passerelle PPP. Certaines permettent de filtrer en fonction du type de données encapsulées dans les datagrammes IP, en fonction des adresses source ou destination présentes dans les en-têtes de ces datagrammes, ou en fonction des ports source ou destination des paquets TCP et UDP encapsulés dans les datagrammes, ce qui permet par exemple de choisir de filtrer par nœuds du réseau ou par services. On peut ainsi transformer certaines passerelles PPP en un maillon important d'une solution firewall de sécurité réseau.

2. Une passerelle PPP doit aussi pouvoir utiliser ses filtres conjointement avec la fonction Dial-on-Demand : certains datagrammes IP ne doivent pas systématiquement initier la connexion PPP. Par exemple, il arrive souvent qu'un utilisateur lance l'application `ping` en boucle, pour obtenir des informations sur l'accessibilité d'un nœud de l'Internet. Les paquets générés, de type ICMP (Internet Control Message Protocol), ne doivent pas être une cause d'activation et de maintien de la ligne par la fonction Dial-on-Demand car un utilisateur qui oublierait par exemple de mettre fin à une commande `ping` après une journée de travail, imposerait à la ligne de transmission une activité toute une nuit, ce qui peut aboutir à une facture téléphonique importante et inutile.

3.2.6 Scripts

Dans le cadre de l'utilisation de PPP sur une connexion intermittente en mode asynchrone, on désigne par script un fichier de configuration permettant à une passerelle PPP d'appeler correctement son homologue chez le fournisseur. En effet, un certain nombre d'opérations est nécessaire à l'établissement d'une connexion PPP asynchrone.

La passerelle doit dialoguer sur le port série en suivant un certain nombre d'étapes :

1. dialogue avec l'ETCD (modem ou adaptateur RNIS, par exemple) afin de l'initialiser et le configurer ; il s'agit souvent d'envoi de commandes Hayes de type `AT` ;
2. envoi d'une demande de connexion en précisant le numéro de téléphone distant (et éventuellement une sous-adresse en RNIS) ; il s'agit là aussi souvent de l'envoi d'une commande Hayes ;
3. dialogue avec le serveur distant pour se connecter sur un compte PPP ;
4. phases PPP :
   - authentification (optionnelle) ;
   - négociation de paramètres (par exemple l'adresse IP de l'équipement qui tente d'établir la connexion) ;
   - phase de production : encapsulation des datagrammes IP pour jouer le rôle de routeur.

Les trois premières phases de la connexion ne font pas intervenir le protocole PPP. Elles sont néanmoins fondamentales et doivent être automatiquement accomplies par la passerelle PPP. La troisième phase consiste pour la passerelle PPP à dialoguer avec un concentrateur de terminaux chez le fournisseur, relié aux modems. Si le concentrateur n'intègre pas la fonction PPP, il faut lui demander de se connecter à un autre équipement : le serveur PPP. Enfin, il faut souvent s'identifier avec un nom de compte ou nom de login, et un mot de passe associé. Certains fournisseurs éliminent cette dernière phase qui est reprise dans la première phase PPP qui suit : la procédure d'authentification. D'autres fournisseurs se passent d'authentification PPP et ne conservent que l'identification par nom de compte et mot-de-passe à la fin du script.

Le script qui permet à la passerelle PPP de dialoguer correctement avec les équipements du fournisseur pour mettre en place la connexion PPP est donc composé d'une série de paires `envoi`/`attente` (`send`/`expect`), où `envoi` désigne une chaîne de caractères à émettre sur la ligne série, et `attente` désigne une chaîne de caractères à scruter en réception avant de passer à la paire suivante.

Prenons donc un exemple typique d'accès PPP composé, chez le fournisseur, de modems reliés à un concentrateur de terminaux, ce dernier étant connecté par le réseau local Ethernet à une machine disposant d'un serveur PPP. Le client se voit attribuer le compte `lucstone` et le mot de passe `elibeurt` sur le serveur PPP. Le concentrateur de terminaux présente une bannière à laquelle il faut répondre par la commande `connect PPP` afin d'accéder au serveur. Ce dernier demande alors le nom de compte et le mot-de-passe, puis commence la session PPP.

Avec un terminal de type VT100, la connexion se traduirait par le dialogue suivant :

```
AT\N3%C3&K3&S&D&W&Y
OK
ATDT0,0123456789
Connect 14400

Terminal Server WTB1023
This access is restricted : no use allowed without permission.

command% connect PPP

     **************
     * PPP Server *
     **************

Welcome on the PPP Server model 6410.

login: lucstone
password: XXXXXXXX
PPPstarting
fno!@#~IDFSF
```

Les caractères de la dernière ligne de l'échange correspondent au début du protocole PPP.

Voici donc la liste des paires `envoi`/`attente` à entrer dans un fichier de configuration de passerelle PPP pour mettre en place cette connexion :

envoi	**attente**
`AT N3%C3&K3&S&D&W&Y \r`	`OK`
`ATDT0,0123456789 \r`	`command%`
`connect PPP \r`	`ogin:`
`lucstone \r`	`word:`
`elibeurt \r`	

Le caractère « `\r` » indique un retour chariot. Dans la deuxième colonne, on préfère scruter les chaînes « ogin: » et « word: » plutôt que « login: » et « password: », car certains fournisseurs

utilisent une majuscule en première lettre, ce qui donne « Login: » au lieu de « login: », et le script peut ne pas le reconnaître. Avec « ogin: » et « word: », on s'adapte à tous les cas de figure.

Pour mettre au point un script de connexion PPP, une technique efficace consiste à se connecter une première fois par un émulateur de terminal, afin d'en déduire, comme nous l'avons fait avec notre exemple, la liste des paires de configuration, puis de les insérer dans les écrans ou fichiers de configuration de la passerelle PPP.

3.2.7 Authentification

Deux systèmes d'authentification, au choix du fournisseur, sont utilisés au sein de PPP ; il s'agit de PAP (Password Authentication Protocol) et CHAP (Challenge Handshake Authentication Protocol). Le fournisseur peut aussi choisir de n'en utiliser aucun.

PAP procède par une identification à l'aide d'un couple nom/mot de passe. Son principe est donc le même que l'identification par mot de passe dans le script de connexion, à la différence près que cette identification se passe ici dans la phase PPP.

CHAP définit quant à lui un secret. Comme le mot de passe, le secret est connu de la passerelle PPP du client et du serveur PPP du fournisseur Internet. Mais à la différence d'un mot-de-passe, le secret n'est pas envoyé au distant sur la ligne de transmission, il est utilisé par le fournisseur pour proposer un *challenge* à la passerelle du client, *challenge* qui ne peut être résolu avec succès que par la connaissance du secret. Ce *challenge* est construit pour l'occasion, il change à chaque demande de connexion. CHAP apporte donc un niveau de sécurité supplémentaire par rapport à PAP : même si la ligne de transmission est écoutée par une personne mal intentionnée, cette dernière ne pourra pas, par la suite, en déduire le secret, elle ne pourra donc pas se connecter en tentant de se faire passer pour la passerelle PPP.

PAP et CHAP permettent au client de s'identifier auprès du fournisseur et peuvent aussi permettre au client d'identifier le fournisseur. Chaque fournisseur possède sa propre politique d'utilisation de PAP, CHAP ou de l'authentification par script. Souvent, ces choix dépendent, chez un même fournisseur, du service auquel on s'abonne. Il faut noter que la plupart des passerelles PPP proposent PAP, mais que beaucoup ne disposent pas encore de CHAP. Il faut donc vérifier que les possibilités d'authentification de la passerelle PPP dont on compte faire l'acquisition sont compatibles avec la solution imposée par le fournisseur.

3.2.8 Négociation

Après la phase d'authentification, la passerelle PPP va négocier un certain nombre de paramètres avec son homologue chez le fournisseur. On notera notamment, parmi les paramètres négociés, les suivants :

– adresse IP de l'interface PPP locale ;

- adresse IP de l'interface PPP du fournisseur ;
- compression des en-têtes de datagrammes IP par la méthode de Van Jacobson.

Traditionnellement, quand on configure une passerelle PPP, on peut choisir, pour chacun des paramètres susceptibles d'être négociés :

- de ne pas fournir de valeur et d'attendre ce que le distant va proposer ;
- de fournir une valeur éventuellement modifiable si le distant en fournit une distincte ;
- de fournir une valeur qui ne sera pas modifiée même en cas de demande de la part du distant.

Il est souvent souhaitable pour le client de laisser le serveur PPP du fournisseur lui imposer les valeurs des paramètres négociables, la configuration n'en est que plus simple.

3.3 Routeurs de proximité

3.3.1 Routeurs CISCO

Dans les sections qui suivent, et plus particulièrement dans le cadre de la connexion par RNIS, ligne spécialisée ou par Transpac, nous allons étudier la configuration d'équipements spécialisés, appelés routeurs de proximité, afin de raccorder un réseau local au réseau du fournisseur Internet. Nous choisissons dans cet ouvrage de décrire différents modèles de la gamme CISCO. Cette société américaine fournit des routeurs souvent utilisés dans le monde de l'Internet et plus particulièrement en France, autant sur les dorsales des fournisseurs que chez leurs clients. D'autre part, le système d'exploitation temps-réel qui les accompagne, et qui propose un grand nombre d'options, est régulièrement remis à jour pour tenir compte des dernières normes tout droit sorties des groupes de travail de l'IETF[4].

3.3.2 Mise en marche d'un routeur et configuration minimale

Avant d'examiner des configurations spécifiques à certains types de raccordements, examinons les caractéristiques générales de ces équipements, en prenant pour exemple un routeur de proximité de la série 2500.

Un routeur CISCO dispose d'une mémoire RAM non volatile (NVRAM) qui contient la configuration chargée au démarrage, à moins que l'utilisateur ne demande à la charger depuis un serveur TFTP (Trivial File Transfer Protocol).

Pour configurer un routeur CISCO pour la première fois, il faut relier un terminal sur son port console, par exemple un micro-ordinateur PC muni d'une interface série et d'un émulateur de terminal. Les paramètres par défaut sont : 9 600 bits/s, 8 bits de données, pas de parité, 1 bit

4. Internet Engineering Task Force

de départ, 2 bits d'arrêt. On met le routeur sous tension et le message suivant apparaît alors à l'écran :

```
System Bootstrap, Version X.X(XXXX) [XXXXX XX], RELEASE SOFTWARE
       Copyright (c) 1986-199X by Cisco Systems
       2500 processor with 4096 Kbytes of main memory
       Notice: NVRAM invalid, possibly due to write erase.
       F3: 5797928+162396+258800 at 0x3000060
       Restricted Rights Legend
       Use, duplication, or disclosure by the Government is
       subject to restrictions as set forth in subparagraph
       (c) of the Commercial Computer Software - Restricted
       Rights clause at FAR sec. 52.227-19 and subparagraph
       (c) (1) (ii) of the Rights in Technical Data and Computer
       Software clause at DFARS sec. 252.227-7013.
       Cisco Systems, Inc.
       170 West Tasman Drive
       San Jose, California 95134-1706
       Cisco Internetwork Operating System Software
       IOS (tm) X000 Software (XXX-X-X), Version XX.X(XXXX) [XXXXX XXX]
       Copyright (c) 1986-199X by Cisco Systems, Inc.
       Compiled Fri 20-Oct-9X 16:02 by XXXXX
       Image text-base: 0x03030FC0, data-base: 0x00001000
       Cisco 25XX (68030) processor (revision A) with 4092K/2048K bytes of memory.
       Processor board ID 00000000
       Bridging software.
       SuperLAT software copyright 1990 by Meridian Technology Corp).
       X.25 software, Version X.X, NET2, BFE and GOSIP compliant.
       TN3270 Emulation software (copyright 1994 by TGV Inc).
       Basic Rate ISDN software, Version X.X.
       X Ethernet/IEEE 802.3 interface.
       2 Serial network interfaces.
       1 ISDN Basic Rate interface.
       32K bytes of non-volatile configuration memory.
       8192K bytes of processor board System flash (Read ONLY)
       Notice: NVRAM invalid, possibly due to write erase.
       --- System Configuration Dialog ---
       At any point you may enter a question mark '?' for help.
       Refer to the 'Getting Started' Guide for additional help.
       Use ctrl-c to abort configuration dialog at any prompt.
       Default settings are in square brackets '[]'.
       Would you like to enter the initial configuration dialog? [yes]:
```

On répond `yes` pour entrer en phase de configuration manuelle.

On voit alors apparaître l'écran suivant :

```
First, would you like to see the current interface summary? [yes]:
       Any interface listed with OK? value "NO" does not have a valid configuration
       Interface IP-Address OK? Method Status Protocol
       Ethernet0 unassigned NO not set up down
       BRI0 unassigned NO not set up up
       Serial0 unassigned NO not set down down
       Serial1 unassigned NO not set down down

Configuring global parameters:
       Enter host name [Router]:
```

À partir de ce point, le système va créer une configuration initiale à partir des réponses aux différentes questions concernant principalement les adresses IP et les protocoles à mettre en

jeu. Par la suite, nous serons amenés à mettre en place des configurations particulières et nous examinerons les configurations complètes à saisir, ce n'est donc pas important si certains des paramètres qu'on entre maintenant sont amenés à changer.

3.3.3 Différents modes d'opération

Un routeur CISCO se pilote depuis un émulateur de terminal ou depuis un terminal virtuel par un accès `PAD` ou `telnet` par exemple.

On distingue cinq modes d'opération qui peuvent être repérés par une invite propre à chacun d'eux. Cette invite dépend en partie du nom du routeur, nous supposerons que ce nom est celui proposé par défaut lors du premier démarrage du système : `Router`.

Les cinq modes en question sont les suivants :

1. Mode utilisateur : l'administrateur est placé dans ce mode lorsqu'il se connecte sur le routeur, après fourniture d'un mot-de-passe. Seulement un sous-ensemble des commandes y est accessible et son invite se termine par le signe « > ».

```
<ls@fenetre> telnet router-cisco.fenetre.fr
Trying 192.168.22.33...
Connected to  192.168.22.33.
Escape character is '^]'.

User Access Verification

Password:
Router>show version
Cisco Internetwork Operating System Software
IOS (tm) 1000 Software (CPA1005-XY-M), Version 11.0(4), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1995 by cisco Systems, Inc.
Compiled Tue 19-Dec-95 03:01 by alanyu
Image text-base: 0x02004000, data-base: 0x02208448

ROM: System Bootstrap, Version 4.14(6)[fc3], SOFTWARE

router-cisco uptime is 9 weeks, 5 days, 9 hours, 51 minutes
System restarted by reload at 12:42:47 METDST Mon Sep 23 1996
System image file is "c2500-i-l_112-0_25.bin", booted via flash
Host configuration file is "router-conf", booted via tftp from 10.0.0.2
Network configuration file is "network-conf", booted via tftp from 10.0.0.2
...
```

2. Mode privilégié : ce mode fournit l'accès à l'ensemble des commandes de maintenance, par exemple pour consulter les états des différentes interfaces ou les tables des protocoles de routage. C'est aussi un passage obligé pour atteindre le mode de configuration. Dans ce mode, l'invite se termine par le signe « # ». Pour y entrer, il faut utiliser la commande `enable`.

   La commande `show interface`, accessible depuis le mode privilégié, permet de consulter les caractéristiques d'une interface quelconque :

```
Router>enable
Password:
Router#show interface ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is QUICC Ethernet, address is 0000.0c33.0251 (bia 0000.0c33.0251)
  Internet address is 192.168.22.33 255.255.255.224
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:09, output 0:00:03, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     852555 packets input, 58923244 bytes, 0 no buffer
     Received 67785 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     758908 packets output, 267156071 bytes, 0 underruns
     0 output errors, 115 collisions, 0 interface resets, 0 restarts
     0 output buffer failures, 0 output buffers swapped out
```

3. Mode de configuration global : il permet d'entrer des commandes qui vont modifier la configuration de l'équipement. Depuis le mode privilégié, on peut charger une configuration par TFTP avec la commande `configure network`, mais on préfère habituellement passer du mode privilégié au mode de configuration depuis le terminal branché sur la console et faire les modifications manuellement. On utilise pour cela la commande `configure terminal` depuis le mode privilégié. L'invite du mode de configuration global est du type « `Router(config)#` » où `Router` est remplacé par le nom du routeur. Définissons par exemple ce nom avec la commande de configuration globale `hostname`. Nous allons constater que cela va ainsi immédiatement changer l'invite. Pour sortir de ce mode, il faut utiliser la touche ^Z (Contrôle+Z), et on revient ainsi au mode privilégié :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname fenetre
fenetre(config)#^Z
fenetre#
```

   Notons que les modifications sont toujours prises en compte immédiatement.

4. Sous-commandes de configuration : certaines commandes entrées en mode de configuration global vont mettre le routeur dans un mode de saisie de sous-commandes dans le contexte de la dernière commande globale. L'invite est alors modifiée et reflète l'action de la commande globale. Modifions par exemple l'adresse IP d'une interface Ethernet. La commande de configuration globale `interface ethernet 0` permet d'entrer dans un mode de saisie de sous-commandes s'appliquant à l'interface Ethernet 0 :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#ip address 192.168.22.33 255.255.255.224
Router(config-if)#^Z
Router#
```

5. Mode Moniteur ROM : ce mode permet un accès de très bas niveau au routeur. Dans ce mode, le routeur est inopérant.

Que faire lorsqu'on a perdu le mot de passe ?

C'est à partir du mode Moniteur ROM qu'on peut reprendre la main sur un routeur dont on a oublié le mot de passe. Pour y entrer, il faut redémarrer l'équipement et enclencher la touche BREAK du terminal branché sur le port console pendant les soixante premières secondes de la procédure d'amorçage.

3.3.4 Opérations de base

Examinons les opérations de base qu'on est amené à effectuer quelle que soit la configuration à mettre en place.

Aide contextuelle

À tout moment, qu'on soit en mode utilisateur, privilégié ou en phase de configuration, on peut obtenir une aide contextuelle en utilisant le caractère « ? ». On remarquera que pour une même commande, `show` par exemple, seuls les arguments autorisés dans le mode courant seront affichés. En voici un exemple :

```
Router>show ?
  clock     Display the system clock
  history   Display the session command history
  hosts     IP domain-name, lookup style, nameservers, and host table
  sessions  Information about Telnet connections
  terminal  Display terminal configuration parameters
  users     Display information about terminal lines
  version   System hardware and software status
Router>show
Router>enable
Password:
Router#show ?
  access-expression  List access expression
  access-lists       List access lists
  accounting         Accounting data for active sessions
  aliases            Display alias commands
  arp                ARP table
  async              Information on terminal lines used as router interfaces
  bridge             Bridge Forwarding/Filtering Database [verbose]
  buffers            Buffer pool statistics
  clock              Display the system clock
  cmns               Connection-Mode networking services (CMNS) information
  compress           Show compression statistics.
  configuration      Contents of Non-Volatile memory
  controllers        Interface controller status
  debugging          State of each debugging option
...
```

Le caractère « ? » en phase de configuration indique le type du prochain argument même si

plus d'un argument est possible. Regardons par exemple comment remplir tour à tour les différents arguments qui complètent la commande de configuration `ip route` permettant de définir des routes statiques :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route ?
  A.B.C.D  Destination Network IP address

Router(config)#ip route 192.168.22.64 ?
  A.B.C.D   Destination mask or forwarding router's address
  Ethernet  IEEE 802.3
  Null      Null interface
  Serial    Serial

Router(config)#ip route 192.168.22.64 255.255.255.224 ?
  <1-255>   Distance metric for this route
  A.B.C.D   Forwarding router's address
  Ethernet  IEEE 802.3
  Null      Null interface
  Serial    Serial
  tag       Set tag for this route
  <cr>

Router(config)#ip route 192.168.22.64 255.255.255.224 192.168.22.36
Router(config)#^Z
Router#
```

Sauvegarde de la configuration

Deux configurations sont en mémoire à tout instant :

- la configuration active ;
- la configuration présente en NVRAM et activée au moment du démarrage (à moins que l'administrateur n'ait choisi un chargement de la configuration par TFTP au démarrage).

Sachant que les commandes de configuration sont prises en compte immédiatement, mais qu'elles ne sont pas placées dans la configuration en NVRAM automatiquement, une erreur de manipulation a rarement des conséquences graves, car, au pire, le routeur devient inopérant et il suffit de le redémarrer pour recharger la configuration correcte enregistrée en NVRAM.

Une fois qu'on a confiance dans la configuration active, on peut la recopier en NVRAM par la commande du mode privilégié `write memory`.

Affichage de la configuration

L'affichage d'une configuration par un routeur CISCO est extrêmement agréable : le routeur se contente d'afficher les commandes qui permettent d'obtenir la configuration en question.

On remarque donc que chez CISCO, le langage de configuration et le langage de description de configuration sont les mêmes, ce qui est très apprécié des administrateurs réseau.

Pour afficher la configuration enregistrée en NVRAM, on dispose de la commande `show configuration`:

```
Router#show configuration
Using 610 out of 7506 bytes
!
version 11.0
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname Router
!
username routeurFournisseur password leSecret
isdn switch-type vn3
!
boot system flash
enable password LucStone
!
...
```

Attention : `show configuration` affiche la configuration en NVRAM, il ne s'agit donc pas de la configuration active. Pour afficher cette dernière, on peut par exemple l'écrire en NVRAM (`write memory`) puis utiliser `show configuration`. Mais pour afficher la configuration courante sans modifier la configuration en NVRAM, on utilise `write terminal`:

```
Router#write terminal
Building configuration...

Current configuration:
!
version 11.0
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname Router
!
username routeurFournisseur password leSecret
isdn switch-type vn3
!
boot system flash
enable password LucStone
!
...
```

Modification d'un paramètre

Lorsqu'on modifie un paramètre de configuration, par exemple l'adresse IP d'une interface, on doit supprimer le paramètre erroné avant de saisir la nouvelle commande de configuration.

Pour cela, il suffit d'écrire la ligne erronée en la faisant précéder du préfixe `no`. Changeons par exemple l'adresse IP de l'interface Ethernet de notre routeur :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#no ip address 192.168.22.33 255.255.255.224
Router(config-if)#ip address 192.168.22.34 255.255.255.224
Router(config-if)#^Z
Router#
```

Redémarrage

Lorsque la configuration semble correcte mais que le routeur ne réagit pas comme on l'attend, il est parfois utile de le redémarrer. On utilise pour cela la commande `reload`.

3.4 Réseau téléphonique commuté

3.4.1 Bande passante

L'ETCD utilisé lors d'un raccordement par le RTC est un équipement appelé modem, dont le rôle est de moduler les signaux numériques correspondant aux informations qui transitent entre le réseau local du client et celui du fournisseur, en les transformant en signaux analogiques pour qu'ils puissent transiter sur une ligne téléphonique.

La débit maximal qu'on peut ainsi atteindre dépend de la largeur de bande disponible, ainsi que du rapport signal/bruit. Une ligne téléphonique possède une bande passante qui commence à 300 Hz et se termine vers 3 400 Hz, la largeur de bande est donc de 3 100 Hz. On définit la rapidité de modulation, qui s'exprime en bauds, comme le nombre maximal de signaux qui peuvent être transmis sur un canal, pendant un intervalle de temps donné. Le théorème de Shannon indique que la rapidité de modulation R, est au plus égale au double de la largeur de bande B :

$$R < 2.B$$

Ainsi, pour une ligne téléphonique, la rapidité de modulation maximale est de 6 200 Bauds.

Chaque signal peut désigner plusieurs états dont le nombre maximal est imposé par le rapport signal/bruit. Shannon a montré que ce nombre maximal n, appelé valence du signal, est

calculable par la formule suivante :

$$n = \sqrt{1 + \frac{Signal}{Bruit}}$$

Le débit en bits/s dépend évidemment de la rapidité de modulation maximale, et du nombre n d'états associés à chaque signal. Le débit D peut ainsi être calculé par la formule :

$$D = R.\log_2(n)$$

En reprenant la première inégalité sur R, et les deux formules qui suivent, on en déduit la relation :

$$D < B.\log_2(1 + \frac{Signal}{Bruit})$$

En appliquant ce résultat aux caractéristiques d'une ligne téléphonique, c'est-à-dire 3 100 Hz de largeur de bande et un rapport signal/bruit en puissance de 100 à 1 000, on obtient une limite théorique maximale comprise entre 20 000 et 30 000 bits/s. Notons qu'avec des techniques de compression de données, on peut dépasser ces débits, mais on reste néanmoins en dessous des débits obtenus avec une liaison RNIS ou une ligne spécialisée. C'est pourquoi les liaisons RTC constituent le type d'accès Internet de plus faible débit, et de coût le plus économique qui soit.

3.4.2 Modems

Caractéristiques techniques

On caractérise les modems par :

- le type de jonction ETTD/ETCD, qui définit des caractéristiques électriques, mécaniques et fonctionnelles, et le protocole d'échange de données numériques de plus bas niveau (échange au niveau bit pour les liaisons synchrones et au niveau caractère pour les liaisons asynchrones) ;
- le mode de transmission, asynchrone ou synchrone ;
- le débit de la jonction ;

- le débit sur la ligne de transmission ;
- la technique de modulation ;
- les techniques de compression de données ;
- les techniques de correction d'erreur.

En ce qui concerne les liaisons Internet par modem, la jonction choisie est dans la grande majorité des cas de type asynchrone. Elle se conforme à l'avis V24 de l'ITU-T.

L'ITU-T est un organisme qui émane d'une entité de l'ONU : l'Union internationale de télécommunication (International Telecommunication Union). L'ITU-T était nommé auparavant Comité consultatif international pour la télégraphie et la téléphonie (CCITT), et son rôle est d'établir des normes internationales dans les domaines des télécommunications.

L'avis V24 qu'il a fait paraître définit la fonction de chaque signal d'une jonction particulière. Les caractéristiques électriques des signaux sont quant à elles définies par l'avis V28 de l'ITU-T. On en trouve deux types de brochages : le connecteur DB-9 à 9 points et le connecteur DB-25 à 25 points. Le plus souvent, le connecteur est de type femelle du côté modem et mâle du côté ETTD. Ce type de jonction est appelé RS-232-C.

La figure 3.4 décrit le connecteur de type DB-9, et le tableau suivant indique la fonction des différents signaux :

Broche	Signal	Fonction	Direction
1	DCD	détection de porteuse	vers l'ETTD
2	RD	réception de données	vers l'ETTD
3	TD	transmission de données	vers le modem
4	DTR	ETTD prêt	vers le modem
5	M	Terre	
6	DSR	modem prêt	vers l'ETTD
7	RTS	demande pour émettre	vers le modem
8	CTS	prêt à émettre	vers l'ETTD
9	XI	indicateur d'appel	vers l'ETTD

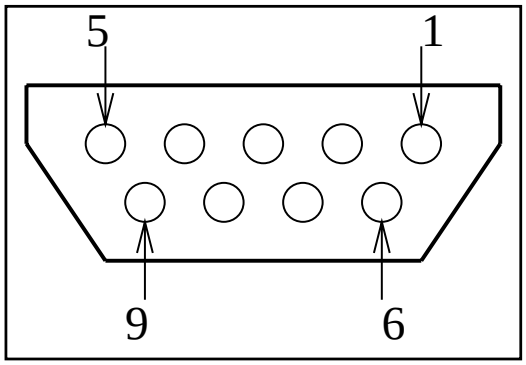

Figure 3.4 *Connecteur DB-9 d'un modem*

La connexion du modem avec la prise téléphonique se fait souvent avec un connecteur de type RJ-11 du côté modem. La figure 3.5 décrit le connecteur de type RJ-11, et le tableau suivant indique la fonction des différents signaux :

Broche	Signal	Fonction
1	T1	combiné point 1
2	L2	ligne point 2
3	L1	ligne point 1
4	T2	combiné point 2

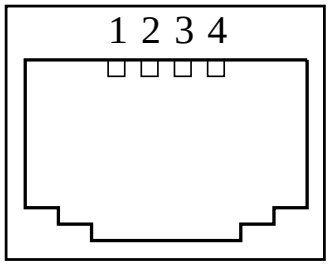

Figure 3.5 *Connecteur RJ-11 d'un modem*

Configuration

Pour configurer un modem, il faut le brancher sur la prise V24/V28 asynchrone d'un ETTD, par exemple un micro-ordinateur ou un terminal de données. Un logiciel de type émulateur de terminal est nécessaire si on utilise un micro-ordinateur.

La configuration de la jonction côté terminal dépend du type d'ETTD. Par exemple, si un émulateur de terminal est utilisé, on trouvera à partir des menus une boîte de dialogue permettant de choisir parmi les différentes options.

La configuration du côté modem se fait à l'aide de commandes normalisées HAYES, aussi appelées commandes AT. Elles doivent être entrées sur le modem par la jonction, depuis le terminal. Entre deux modems de marque différente, ces commandes peuvent changer. Les exemples de commandes AT que nous allons fournir par la suite sont compatibles avec les modems de la marque OLITEC. Conformez-vous à la documentation fournie avec votre modem pour vérifier que les commandes sont les mêmes, ou pour trouver leur équivalent.

Le débit sur la jonction asynchrone doit tout d'abord être configuré sur le terminal. Il est conseillé d'utiliser un débit supérieur à celui sur la ligne de transmission, car souvent un protocole de compression y est mis en place. On ne pourrait pas, ainsi, en profiter si la jonction avait le même débit que la ligne de transmission, car il n'y a pas de compression sur la jonction. Les débits de jonction qu'on retrouve le plus souvent sont : 300 bits/s, 1200 bits/s, 2400 bits/s, 4800 bits/s, 9600 bits/s, 14400 bits/s, 38400 bits/s, 57600 bits/s et 115200 bits/s. Ce débit est imposé par l'ETTD, et reconnu automatiquement par le modem lorsque l'utilisateur

envoie la première commande de configuration. La commande la plus simple, `AT`, permet de vérifier que la jonction est opérationnelle. Dès sa réception, le modem doit renvoyer le code de retour `OK` :

```
AT
OK
```

Si les caractères saisis n'apparaissent pas à l'écran, on utilise la commande `ATE1` pour passer en mode echo.

Le format des données, définissant le nombre de bits d'information, de bits de départ (bits de START), de bits d'arrêt (bits de STOP) et la parité, doit aussi être défini. Avec les protocoles utilisés pour la connexion Internet par modem (SLIP, PPP ou UUCP), on va généralement choisir la configuration suivante : 8 bits de données, 1 bit de départ, 1 bit d'arrêt, pas de parité. Cette configuration doit être mise en place sur le terminal, et est détectée par le modem lors de la première commande AT.

Le débit et les protocoles de transmissions sur la ligne sont définis par un certain nombre d'avis de l'ITU-T. Pour certains de ces avis, plusieurs débits sont disponibles ; nous fournissons ici les avis les plus utilisés, et, pour chacun d'eux, le débit communément adopté, ainsi que le code AT correspondant :

Avis	**Débit**	**commande AT**
V21	300 bits/s	B7
V22	1 200 bits/s	B6
V22bis	2 400 bits/s	B8
V23	1 200 bits/s en réception, 75 bits/s en émission	B2
V32	9 600 bits/s	B9
V32bis	14 400 bits/s	B10
V34	28 800 bits/s	B20
V34bis	33 600 bits/s	consulter la documentation du modem

Ainsi, pour configurer un débit de ligne de 28 800 bits/s en V34, il suffit d'entrer la commande suivante :

```
ATB20
OK
```

Il se peut qu'au cours de la transmission, ou même au début de l'échange, la qualité de la ligne ou les possibilités du modem distant ne soient pas suffisantes pour dialoguer au débit défini par le modem local. Une nouvelle vitesse moins importante peut alors être utilisée par les deux modems : il s'agit de l'opération de repliement. Il faut l'autoriser dans le modem par la commande `%E`, sinon, en cas de dégradation de ligne, il risque de couper la connexion.

Pour éviter le coût d'un délai supplémentaire, il faut passer le modem dans un mode sans tampon, par la commande `\N1`. On ne dispose alors d'aucune compression, il faut que la jonction ait la même vitesse que la ligne. Plusieurs autres possibilités s'offrent à nous. On peut choisir d'utiliser le tampon du modem sans compression ni correction (commande `\N0`), la jonction peut ainsi être plus rapide que la ligne. On peut aussi choisir d'utiliser le tampon et un protocole de correction d'erreur. Il existe deux protocoles de correction d'erreur : V42 et MNP4. On peut imposer V42 (commande `\N4`), MNP4 (commande `\N5`), ou demander au modem de le négocier (commande `\N3`). La correction V42 est à préférer à MNP4. On choisit généralement de demander au modem de négocier :

```
AT\N3
OK
```

De même que pour les protocoles de correction d'erreur, il existe deux protocoles de compression de données sur la ligne de transmission : MNP5 et V42bis. On a donc le choix de ne pas en utiliser (commande `%C0`), d'imposer l'un ou l'autre (commande `%C1` pour MNP5, `%C2` pour V42bis), ou de demander au modem de négocier (commande `%C3`). La compression V42bis est à préférer à MNP5. On choisit généralement de demander au modem de négocier :

```
AT%C3
OK
```

Il existe deux types de contrôles de flux : le contrôle matériel ou *hardware flow control* (commande `&K3`) et le contrôle logiciel ou *software flow control* (commande `&K4`). Le contrôle de flux doit être configuré sur l'ETTD, par exemple dans un menu d'un émulateur de terminal, et sur l'ETCD avec une commande AT. Le contrôle de flux logiciel consiste à utiliser des caractères particuliers permettant à l'ETTD et à l'ETCD de demander un arrêt d'émission de données, par exemple quand leurs tampons sont pleins. Le caractère 19 (^S ou XON) correspond à une demande d'arrêt d'émission, et le caractère 17 (^Q ou XOFF) correspond à une demande de reprise. Cela implique que ces caractères sont réservés et ne peuvent pas traverser la ligne de transmission. C'est souvent une cause de problèmes importants avec les protocoles utilisés pour la connexion Internet. Il faut donc préférer le contrôle de flux matériel, qui utilise les signaux RTS et CTS de l'interface V24/V28. Parfois, les deux types de contrôles de flux sont disponibles mais ce mode de fonctionnement est souvent déconseillé.

Certains câbles V24/V28 sont livrés sans le câblage des signaux RTS ni CTS. C'est parfois le cas dans le monde Macintosh. Il faut alors utiliser un contrôle de flux logiciel, mais certains programmes ne le supporteront pas, ou bien demanderont une configuration particulière (c'est le cas notamment de PPP : on peut lui configurer une *asynchronous-map*, table indiquant les caractères à éviter ; il faut alors y inclure ^S et ^Q si on utilise un contrôle de flux logiciel).

Que faire si la jonction ne transporte pas RTS ni CTS ?

Dans le cas de l'utilisation d'un logiciel incompatible avec le contrôle de flux logiciel et d'un câble de jonction ne rendant pas disponible RTS ni CTS, il faut procéder en quatre étapes :

1. **supprimer toute forme de contrôle de flux (commande `&K`), puisque le contrôle logiciel est incompatible avec le programme utilisé et que le contrôle matériel n'est pas disponible sur le câble de jonction ;**
2. **utiliser une vitesse de jonction identique à la vitesse sur la ligne de transmission afin que les données dans le sens ETTD vers modem ne provoquent pas un engorgement qui amènerait à des pertes de caractères émis par l'ETTD ;**
3. **passer en mode avec tampon sans compression, car la vitesse de jonction est égale à celle de la ligne. Une compression de données de la part du modem distant pourrait alors entraîner un engorgement dans le sens modem local vers ETTD, et l'absence de contrôle de flux amènerait ainsi à des pertes de caractères dans ce sens ;**
4. **activer un contrôle d'erreur.**

Certains ETTD ont un comportement dépendant de l'état du signal DSR de la jonction, qui est normalement activé par le modem uniquement pendant une connexion distante. Par exemple, l'accès à la jonction série sous Unix est souvent impossible depuis un périphérique en mode caractère comme `/dev/tty0`[5] lorsque ce signal est à l'état bas. Pour y remédier, on utilise le périphérique associé `/dev/cua0` qui n'a généralement pas cette limitation.

Il est aussi possible de changer ce comportement de */dev/tty0*. Pour cela, on se connecte à l'aide d'un émulateur de terminal sur */dev/cua0* et on demande alors au modem de systématiquement activer le signal DSR par la commande `&S`.

Le signal DTR est positionné par l'ETTD. Souvent, lorsqu'on redémarre un ETTD, ou à plus forte raison lorsqu'on le met hors tension, ce signal tombe. Le modem peut être configuré pour ignorer le signal DTR (commande `&D`) ou bien pour couper la connexion distante lorsque le signal DTR tombe (commande `&D2`). On est parfois amené à utiliser cette commande par mesure de sécurité, pour s'assurer que la ligne est bien déconnectée lorsque l'ETTD est arrêté, par exemple pour éviter une facture téléphonique excessive due à une connexion maintenue par erreur.

Certains modems, par exemple certains modèles de la marque USRobotics, ont une configuration dépendant de commutateurs de type *on/off* accessibles sur une de leurs faces. Il faut dans ce cas se reporter à la documentation du modem.

Une fois la configuration choisie, il faut la mettre en place dans le modem, l'enregistrer dans un des bancs de registres en mémoire non volatile (commande `&W` pour le banc 0), et enfin indiquer au modem le banc de registres contenant la configuration à charger à la mise sous tension (commande `&Y` pour le banc 0).

5. TTY : Teletype

Notons que plutôt que de les saisir une par une, on peut regrouper les commandes AT en les faisant précéder du préfixe AT suivi de la liste de toutes les commandes choisies.

Voici un exemple de configuration adaptée aux protocoles d'accès Internet sur modem : utilisation d'un tampon, contrôle d'erreur négocié, compression négociée, contrôle de flux matériel, forcer DSR, ignorer DTR, enregistrement dans le banc 0, rappel de ce banc à la mise sous tension.

```
AT
OK
AT\N3%C3&K3&S&D&W&Y
OK
```

3.4.3 UUCP : Unix to Unix Copy

UUCP est un protocole d'interconnexion de machines Unix, à l'origine destiné aux lignes série asynchrones (on peut maintenant encapsuler UUCP sur TCP/IP mais c'est rarement utilisé). Il est donc tout à fait adapté au transport sur RTC par une paire de modems. Il propose une fonction de base qui permet de transférer des fichiers entre machines, et d'exécuter des commandes à distance prenant pour paramètres les fichiers transférés. Notons qu'UUCP ne s'appuie généralement pas sur IP, il se distingue donc de l'Internet. Mais, néanmoins, deux services de cette fonction de base existent aussi sur l'Internet : la messagerie ainsi que le transport d'articles de News (Forums).

Certains fournisseurs proposent donc des accès uniquement à ces deux services Internet par le protocole UUCP sur RTC, à des coûts souvent inférieurs aux connexions Internet complètes.

UUCP est natif sur la plupart des systèmes Unix. Il en existe de plus une réalisation en domaine public appelée Tailor UUCP. Dans le monde PC (DOS, Windows et OS2), le produit UUPC propose le protocole UUCP.

Le principe d'UUCP est d'appeler régulièrement un serveur du même type chez le fournisseur (par exemple de manière automatisée, toutes les heures durant les horaires de travail), d'envoyer les messages et les articles de News générés sur le réseau local et mis en file d'attente depuis le dernier appel, et enfin de récupérer les messages et articles de News à destination du client et qui avaient été mis en file d'attente chez le fournisseur depuis le dernier appel.

On voit donc qu'un délai dépendant de la fréquence de connexion est imposé à la réception comme à l'émission de messages. De plus, seule la messagerie et les forums étant accessibles, il faut utiliser des passerelles *email* pour accéder à des serveurs FTP ou WWW, par exemple. Ce type de passerelle est accessible par courrier électronique : il s'agit d'une adresse *email* dont les messages sont consultés par un automate qui y interprète des requêtes d'accès à des documents disponibles par un protocole tel que FTP. L'automate est mis en place sur une machine connectée à l'Internet, il peut donc aller chercher le document et le retourner à l'émetteur de la requête, encodé dans un *mail*. Il faut néanmoins noter qu'accéder à FTP ou WWW par ce type de passerelle n'est pas franchement ergonomique.

3.4.4 SLIP : Serial Link Internet Protocol

SLIP est un protocole permettant d'encapsuler des paquets IP sur une ligne série. Il dispose de très peu d'options, tant en termes de sécurité qu'en termes de contrôle de la connexion, et s'il a été assez utilisé il y a quelques années, il est maintenant abandonné au profit de PPP, protocole beaucoup plus riche. La totalité des fournisseurs Internet qui proposaient jadis des connexions SLIP fournissent maintenant des accès PPP ; certains fournisseurs proposent les deux types d'accès. Il est conseillé, lorsque cela est possible, d'utiliser PPP en lieu et place de SLIP.

3.4.5 PPP : Point to Point Protocol

Les équipements en jeu

Même s'il est possible d'utiliser PPP sur des connexions par modem en mode série synchrone, les fournisseurs Internet proposent uniquement, dans le cadre des connexions RTC, des accès PPP asynchrones (les trames de plus bas niveau de PPP ne sont pas du même type si la jonction est synchrone ou asynchrone, l'échappement des caractères de synchronisation ne se faisant notamment pas de la même manière).

La figure 3.6 page suivante présente les différents équipements qui entrent en jeu lors d'une connexion PPP par modem :

- Chez le client, on trouve un équipement muni d'un modem, et branché sur un quelconque brin du réseau local, par une carte Ethernet dans notre exemple. Cet équipement fait alors office de passerelle PPP. Le modem est relié à une ligne téléphonique d'un opérateur télécom.
- Le fournisseur possède quant à lui un autocommutateur ou PABX (Private Automatic Branch Exchange) numérique raccordé à l'opérateur télécom. Cet autocommutateur distribue un numéro téléphonique unique vers un groupement de modems suivant un accès cyclique, dit *rotary*. Ainsi, à chaque nouvel appel sur le numéro téléphonique du groupement, il recherche le modem libre suivant. Les modems sont branchés sur un concentrateur de terminaux par des jonctions V24/V28 asynchrones. Le concentrateur est relié au réseau local du fournisseur, par exemple par une interface Ethernet. Il joue alors le rôle de serveur PPP et c'est lui qui va identifier les clients lors de leurs appels, négocier les paramètres de la connexion et permettre le transport des datagrammes IP sur la ligne série. C'est lui aussi qui va permettre d'enregistrer les informations de facturation : temps de connexion, volume de transfert effectué, etc.

  Il existe bien sûr plusieurs variantes du modèle qu'on vient de présenter. Parfois, un seul équipement joue le rôle de l'autocommutateur, des modems et du concentrateur de terminaux, tout en utilisant le protocole PPP. La société ASCEND propose notamment de tels équipements. Parfois, le concentrateur de terminaux ne dispose pas la fonction PPP, il se contente alors de mettre en relation le flux de caractères provenant de la

jonction avec un serveur Unix présent sur le réseau local du fournisseur et qui possède un logiciel serveur PPP.

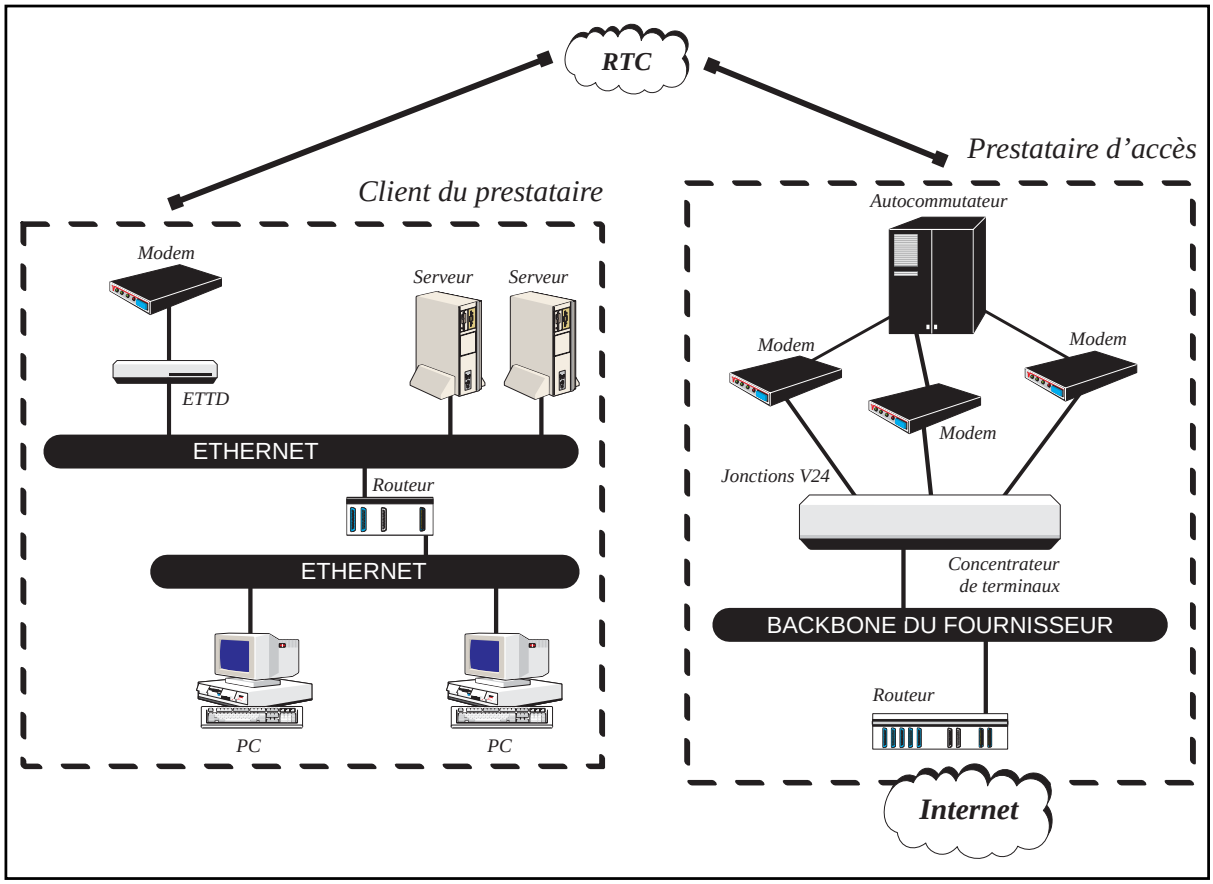

Figure 3.6 *Connexion PPP avec modem : topologie*

Plan d'adressage

Il faut maintenant nous intéresser à la mise en place des adresses des équipements qui entrent en jeu dans la connexion PPP. Reprenons pour l'occasion l'exemple de réseau utilisé au chapitre 2 pour présenter les masques de sous-réseaux.

Rappelons que ce réseau possède un masque de sous-réseaux de valeur `255.255.255.224`, et trois brins Ethernet répartis sur les trois premiers sous-réseaux utilisables dans le réseau de classe C `192.168.22.0`. Le routeur connecté à l'Internet possède une interface Ethernet d'adresse `192.168.22.33` sur le réseau local.

Un modem est donc ajouté à ce routeur, et un logiciel PPP le transforme en passerelle PPP. Il possède une interface en plus de son accès Ethernet : il s'agit de l'interface de la jonction qui le raccorde au modem.

Il existe deux types d'interfaces :

- Les interfaces multipoints (IPTMP : Interface Point To Multipoints) : une interface de ce type raccorde une machine à un support partagé par plusieurs autres équipements. Il peut s'agir par exemple d'un câble Ethernet ou d'un anneau FDDI. À ce type d'interface, on associe les paramètres suivants, qui doivent être configurés manuellement ou automatiquement, par exemple par le logiciel PPP à la suite d'une négociation avec son homologue chez le fournisseur :
  - une adresse IP,
  - un masque de sous-réseaux,
  - une adresse de sous-réseaux,
  - une adresse de diffusion.

  Comme nous l'avons vu au chapitre 2, l'adresse de sous-réseaux et l'adresse de diffusion se déduisent de l'adresse IP et du masque de sous-réseaux.
- Les interfaces point à point (IPTP : Interface Point To Point) : une interface de ce type raccorde une machine à un support partagé avec un unique autre équipement, par exemple une ligne série. À ce type d'interface, on associe les paramètres suivants :
  - une adresse IP locale,
  - l'adresse IP de l'autre équipement sur le support.

Notons qu'il n'y a pas de notion de numéro de sous-réseau, de masque de sous-réseaux, ni d'adresse de diffusion avec ce type d'interface : alors qu'avec une interface multipoint les différents équipements doivent posséder des adresses appartenant à un même sous-réseau d'une même classe, il n'en est rien pour les deux équipements connectés par deux interfaces point à point. De plus, l'adresse IP de l'interface point à point peut être identique à une adresse IP d'une autre interface du même équipement. Cela permet d'économiser des adresses et de simplifier la configuration ; c'est ce qui est utilisé dans notre exemple de plan d'adressage sur la figure 3.7 page ci-contre.

Le routeur du client possède donc deux interfaces qui ont les caractéristiques suivantes :

Nom	**Type**	**Caractéristiques**	
ether0 (Ethernet)	multipoint	adresse IP :	`192.168.22.33`
		réseau :	`192.168.22.0`
		sous-réseau :	`192.168.22.32`
		diffusion :	`192.168.22.63`
		masque de sous-réseaux :	`255.255.255.224`
ppp0 (PPP)	point à point	adresse IP source :	`192.168.22.33`
		adresse IP destination :	`192.168.100.1`

Mise en place du routage

Le fournisseur se charge d'annoncer le réseau de classe C sur l'Internet par le protocole BGP-4. Les paquets à destination du client sont ainsi attirés vers le réseau du fournisseur.

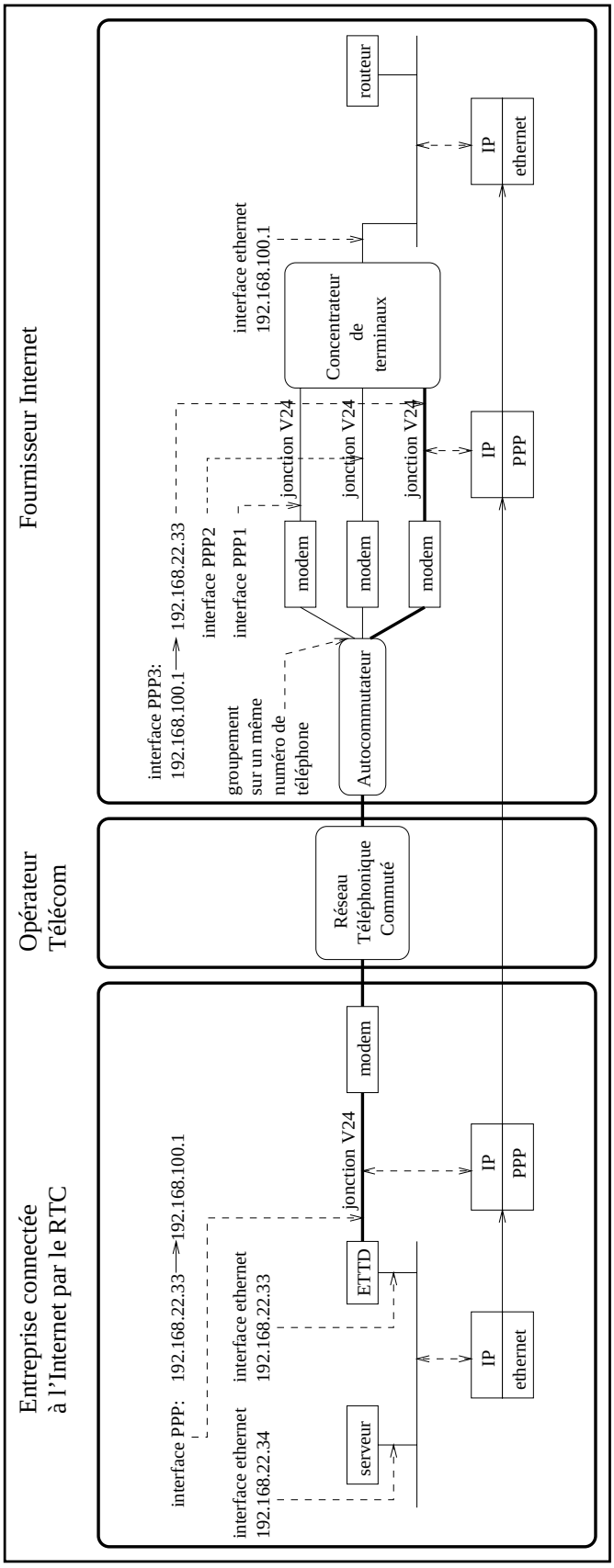

Figure 3.7 *Connexion PPP avec modem : plan d'adressage*

Il met, de plus, en place un routage interne afin que les paquets qui arrivent chez lui soient correctement acheminés vers son client.

Le client doit mettre en place un routage permettant aux paquets à destination de l'Internet de quitter son réseau local à travers l'interface PPP et d'entrer ainsi chez le fournisseur.

La passerelle PPP doit posséder une route par défaut vers son homologue chez le fournisseur, c'est-à-dire vers `192.168.100.1`. Il existe deux possibilités pour ajouter cette route :

- Soit le fournisseur utilise un protocole de routage avec ses clients pour leur annoncer les routes de ses autres clients ainsi que la route par défaut. Dans ce cas, il suffit de mettre en place sur la passerelle PPP le protocole de routage utilisé par le fournisseur et la route par défaut apparaîtra automatiquement.
- Soit le fournisseur n'utilise pas de protocole de routage avec ses clients. Il faut alors rajouter une route par défaut statique sur la passerelle PPP, c'est le cas le plus répandu.

Les autres équipements du réseau doivent posséder une route par défaut vers l'interface Ethernet de la passerelle PPP, c'est-à-dire vers `192.168.22.33`. Il existe deux possibilités pour ajouter cette route :

- soit le client utilise un protocole de routage interne à son réseau, et alors la route par défaut est automatiquement annoncée par la passerelle PPP aux autres équipements ;
- soit le client n'utilise pas de protocole de routage interne, et il lui faut alors configurer manuellement dans chaque équipement la route par défaut en question.

Particularités de la connexion RTC

La connexion RTC est une connexion intermittente. Le client peut ainsi profiter pleinement de tous les services Internet, mais il lui est impossible d'héberger un quelconque serveur (serveur WWW, FTP, DNS, etc.).

En effet, si le client n'est pas connecté et qu'un utilisateur de l'Internet tente d'accéder à un serveur installé sur son réseau, les paquets vont arriver chez le fournisseur, mais ce dernier ne configure jamais son serveur PPP en mode Dial-on-Demand pour rappeler son client, les paquets sont donc perdus dans le réseau du fournisseur.

D'autre part, lorsque le client n'est pas connecté, le courrier est mis en attente chez le fournisseur et c'est au moment où le client se connecte que les messages lui sont redistribués. Ainsi, il faut mettre en place une connexion régulière automatique, simplement pour recevoir les nouveaux messages.

Même si on peut considérer ces particularités comme des défauts de ce type de raccordement, il faut aussi remarquer que cela constitue néanmoins une forme de sécurité non négligeable par rapport aux connexions permanentes. En effet, la connexion RTC rend impossible un éventuel piratage le week-end ou pendant les congés, périodes durant lesquelles les tentatives de piratage se produisent.

PPP sur Macintosh

Sur Macintosh, on utilise le plus souvent MacPPP. Utilisé conjointement avec MacTCP ou OpenTransport[6], il permet à un Macintosh d'utiliser les services de l'Internet, mais il ne fait pas office de passerelle PPP. Cette solution ne permet donc pas de connecter, par l'intermédiaire d'un Macintosh, tout un réseau local. À l'aide de routeurs logiciels, il est toutefois possible de transformer un Macintosh en passerelle PPP, mais cette solution est assez peu utilisée pour le raccordement d'un réseau local dans le cadre d'une activité professionnelle. On lui préfère très souvent un routeur spécialisé, une station de travail sous Unix ou parfois un PC sous Unix, Windows 3.11, Windows 95 ou Windows NT.

PPP sous Windows, Windows 95 et Windows NT

Comme on l'a déjà dit, Windows 95 et Windows NT possèdent une pile TCP/IP. Les versions précédentes n'en possédaient pas, c'est pourquoi de nombreux éditeurs de logiciels en fournissent. Ces dernières ont maintenant toutes été portées sous Windows 95 et NT. Elles comportent souvent beaucoup plus d'outils et de fonctionnalités que la pile native, c'est pourquoi il n'est pas inutile de s'en procurer. Par exemple, les produits Chameleon de Netmanage et OnNet32 de FTP Software possèdent les fonctionnalités Dial-on-Demand et sont accompagnés d'un ensemble de logiciels permettant d'accéder par exemple aux forums, ou de gérer la messagerie, en plus de la fonctionnalité PPP.

Attachons-nous à examiner la configuration de Chameleon de Netmanage pour transformer Windows en passerelle PPP. Nous allons pour cela nous conformer aux huit étapes suivantes de configuration à partir de l'application Custom :

1. Configurons une interface Ethernet comme nous l'avons fait section 2.7.1 page 56.
2. Créons maintenant, par le même procédé, une interface PPP, en plus de l'interface Ethernet.
3. L'onglet ***[Setup/Configuration/Gateway]*** permet de définir une route par défaut (figure 3.8 page suivante).
4. À l'aide de l'onglet ***[Setup/Configuration/Port]***, on configure le port de communication série comme suit (figure 3.9 page suivante) : 57600 bits/s, 8 bits de données, 1 bit d'arrêt, pas de parité, contrôle de flux matériel.
5. À l'aide de l'onglet ***[Setup/Configuration/Dial]***, on définit le numéro d'appel ainsi que différentes options (figure 3.10 page 95) : Dial-on-Demand, déconnexion après une temporisation, reconnexion en cas de perte de ligne accidentelle, etc.
6. À l'aide de l'onglet ***[Setup/Configuration/Modem]***, on définit les caractéristiques du modem (figure 3.11 page 95). Chameleon en déduit automatiquement la chaîne d'initialisation.

6. OpenTransport est disponible à partir du système 7.5.2 ou 7.5.3 selon les modèles de Macintosh.

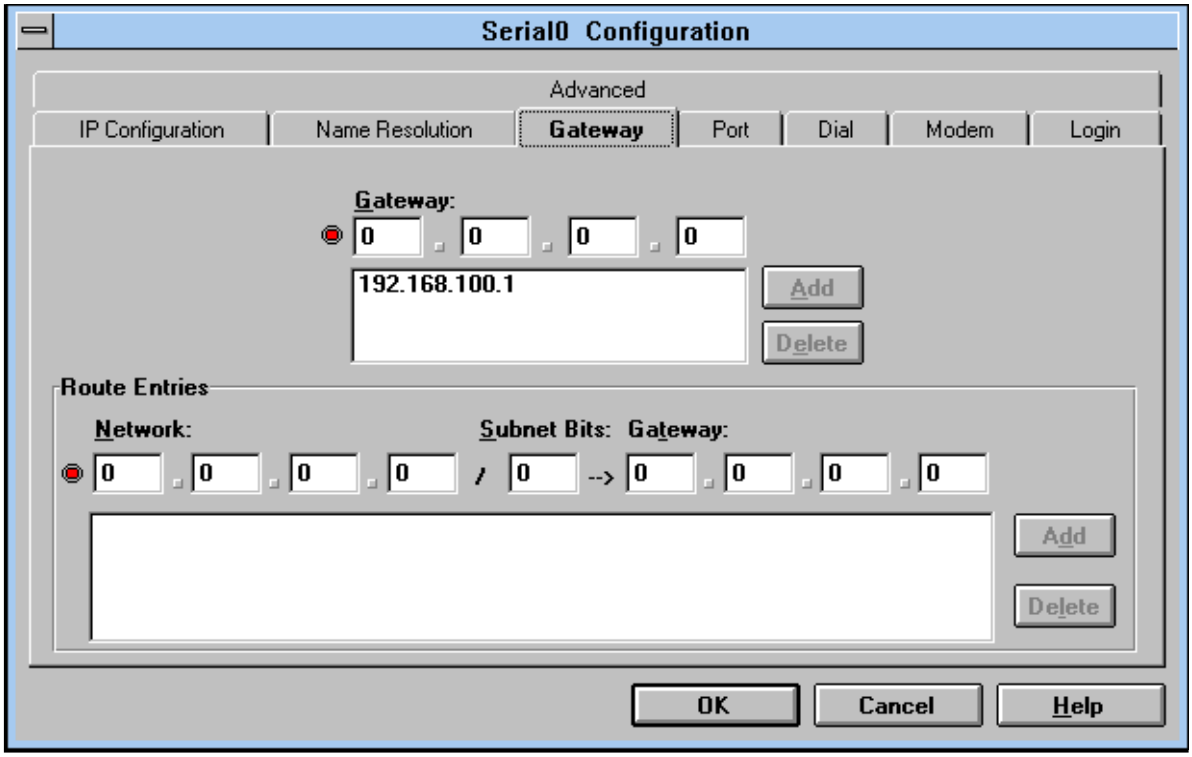

Figure 3.8 *Configuration du routage*

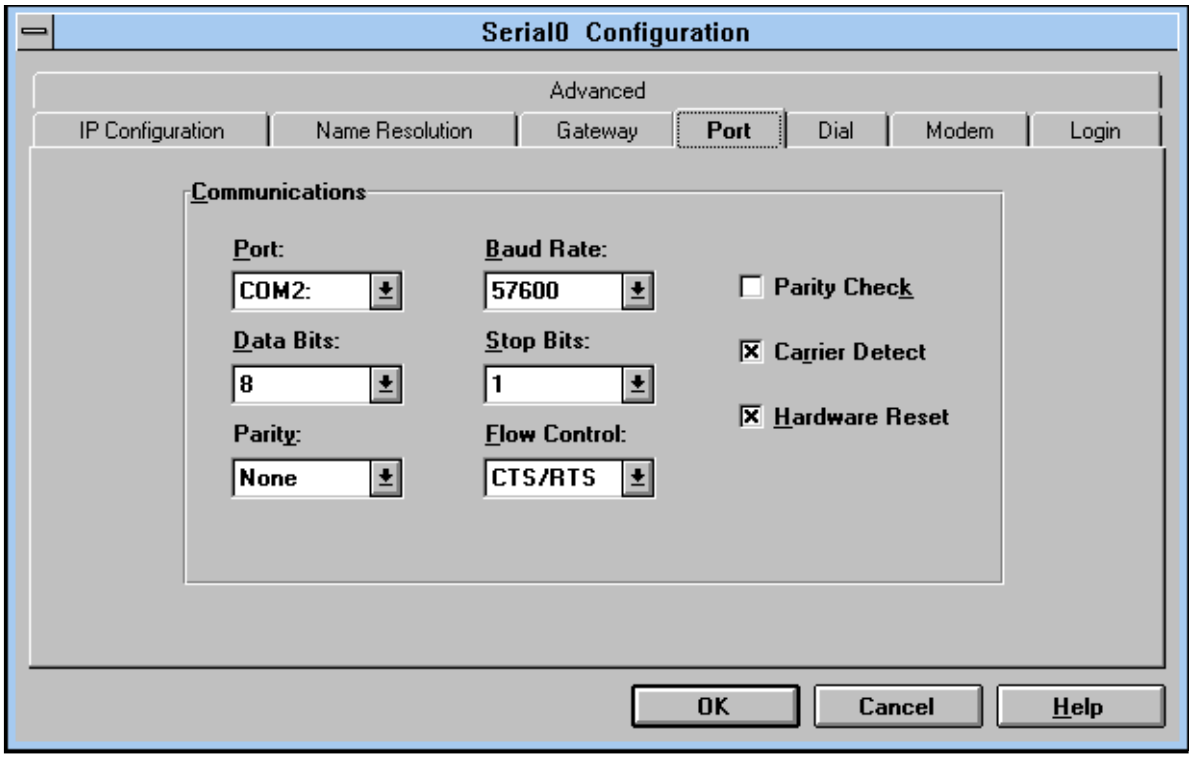

Figure 3.9 *Configuration de l'interface série*

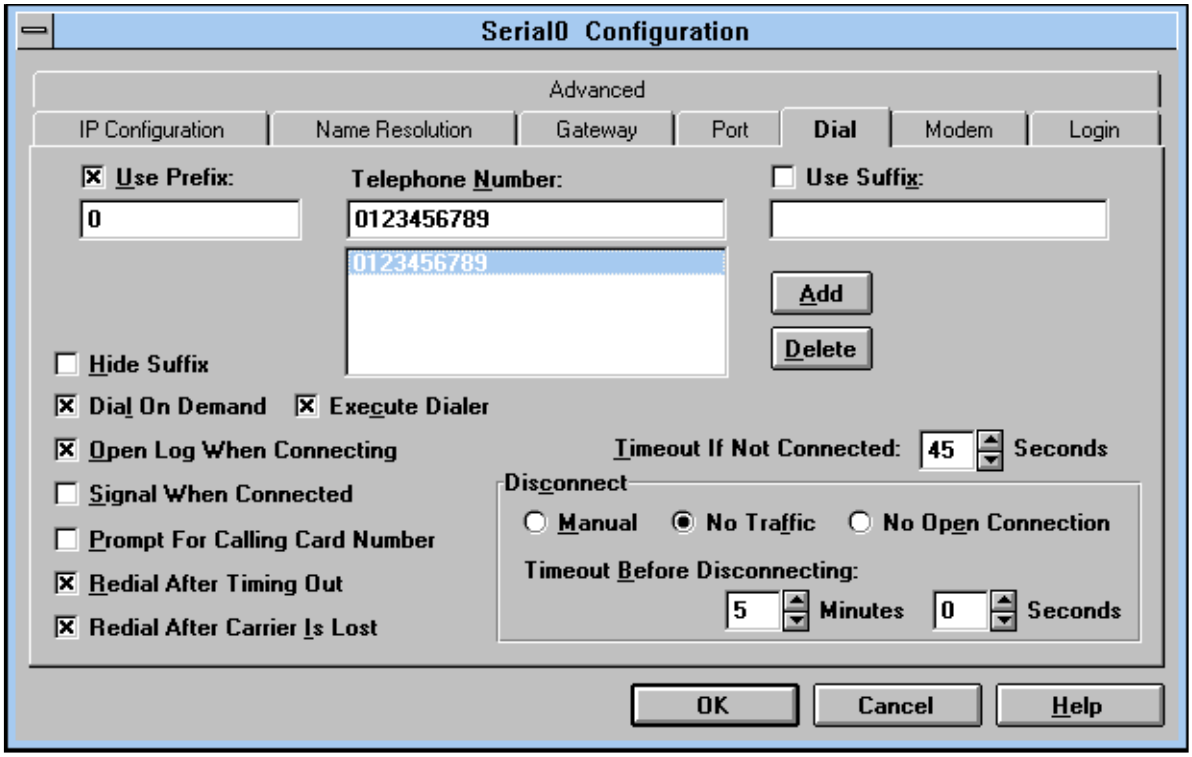

Figure 3.10 *Configuration du contrôle de la ligne*

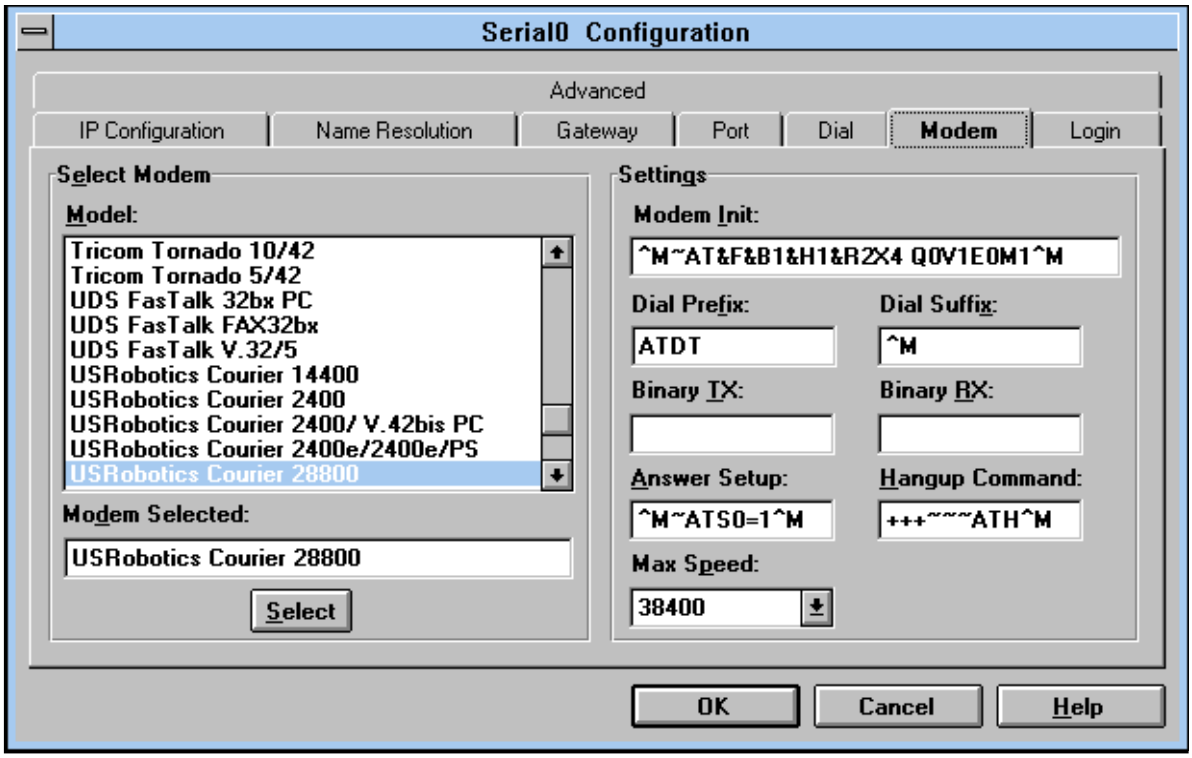

Figure 3.11 *Configuration du modem*

7. À l'aide de l'onglet ***[Setup/Configuration/Login]***, on définit le dialogue entre le PC et le site distant pour passer la phase d'identification de type *Login/Password* (figure 3.12).

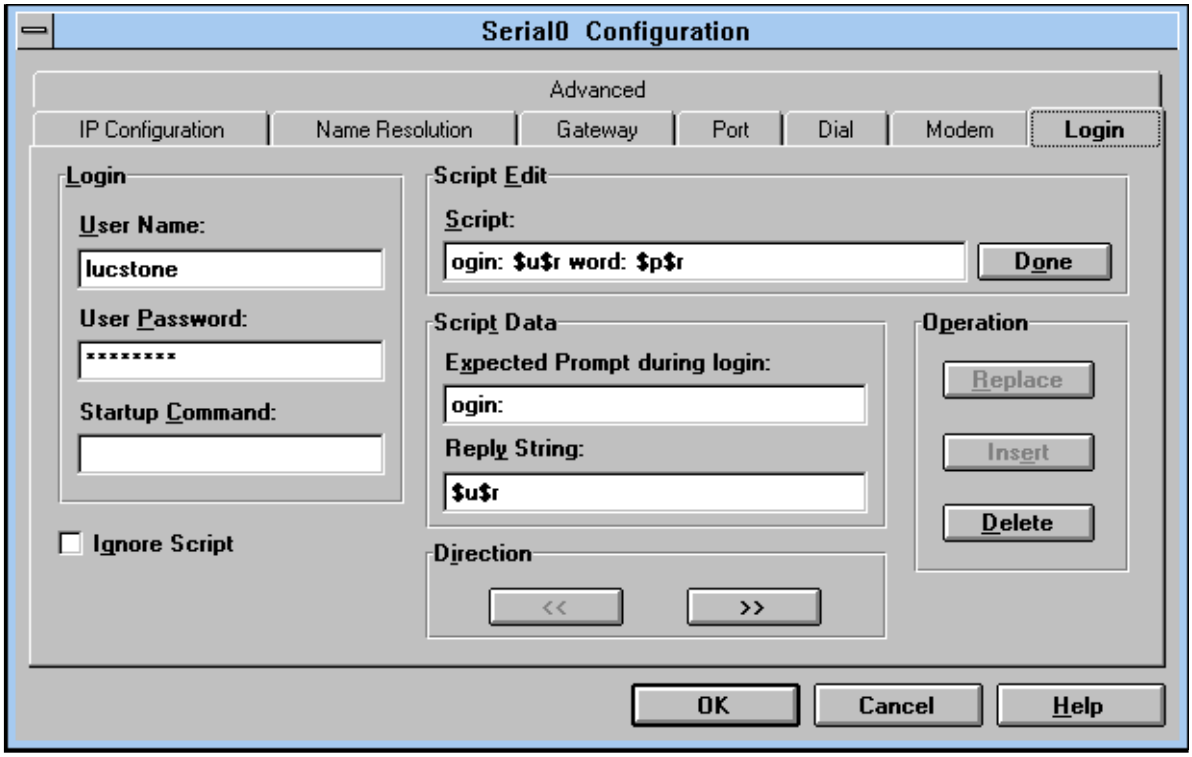

Figure 3.12 *Dialogue d'identification*

8. Enfin, on vérifie que nos deux interfaces sont correctement configurées en revenant au premier écran de l'application Custom (figure 3.13 page suivante). On remarquera qu'on n'a pas eu besoin de configurer l'adresse IP de l'interface PPP, c'est le serveur distant qui l'indiquera au moment de la connexion.

PPP sous Unix

Le monde Unix est particulièrement bien adapté à résoudre les problèmes de raccordement Internet. Certains systèmes Unix sont équipés d'une passerelle PPP en natif, c'est notamment le cas de Solaris de Sun Microsystems. Ce dernier fournit le Dial-on-Demand et les méthodes d'authentification PAP et CHAP.

On trouve des passerelles PPP en domaine public, dont les sources sont disponibles. Il y a par exemple PPP-2.2[7], écrit par Paul MACKERRAS du département d'informatique de l'Australian National University, qui fonctionne sous SunOS, Solaris, NetBSD, Ultrix, Linux, OSF/1, AIX et NeXTStep, et qui reconnaît PAP et CHAP. Malheureusement, il ne propose pas la

7. `ftp://ftp.ibp.fr/pub/networking/ppp/unix`

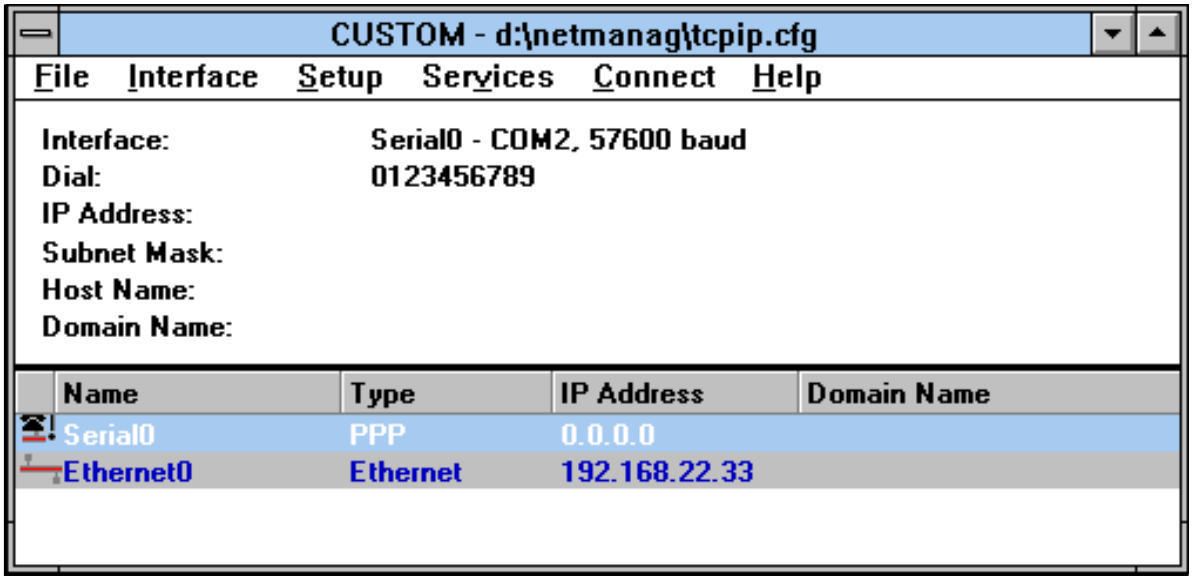

Figure 3.13 *Application Custom : configuration de deux interfaces*

fonction Dial-on-Demand dans la version disponible actuellement. La prochaine version devrait l'intégrer et on peut utiliser pour l'instant l'utilitaire `diald` dont c'est la seule fonction. De nombreuses informations sur `diald` sont disponibles à l'URL suivant :
`http://www.cs.utoronto.ca/ schenk/diald.html`

On peut aussi citer l'excellent produit commercial MorningStar PPP qui fonctionne notamment sous SunOS (stations Sparc et Sun3), Solaris (PC x86 et stations Sparc), Ultrix (DECstations), NeXTStep (stations NeXT), IRIX (stations SGI), AIX (stations RS/6000), HP-UX (stations HP), SCO Xenix (PC x86), BSDI (PC x86). Il dispose notamment de la fonction Dial-on-Demand, de l'authentification PAP ou CHAP, et de possibilités de filtrages complètes.

L'URL `http://www.morningstar.com` permet de recueillir des informations commerciales sur ce produit. On peut aussi utiliser le courrier électronique à destination de `support@morningstar.com`. Sur `ftp://ftp.morningstar.com`, des informations techniques sont disponibles.

Étudions donc la configuration de MorningStar PPP.

L'installation du produit consiste à désarchiver une arborescence de fichiers dans le répertoire `/etc/ppp` (`/usr/lib/ppp` sur SCO).

Il faut adapter cinq fichiers de configuration avant de pouvoir utiliser PPP :

1. `/etc/ppp/Systems` - ce fichier indique un ensemble d'informations sur le système distant qu'on va contacter. Il peut y avoir plusieurs systèmes distants, et donc plusieurs lignes dans ce fichier comme dans les autres que nous allons examiner par la suite, mais nous nous contenterons de nous placer dans le cadre habituel de l'accès à un fournisseur Internet unique.

Le format des lignes contenues dans ce fichier est le suivant :

```
name when device speed phone-number chat-script
```

Voici la signification de ces paramètres :

- name : le nom du système distant ;
- when : une chaîne indiquant les moments où l'on peut contacter cesystème. `Any` indique que le système est joignable à tout moment ;
- device : le nom du périphérique contenu dans `/dev` auquel est relié le modem. Si la valeur de ce paramètre est ACU, les périphériques indiqués dans le fichier `/etc/ppp/Devices` dont le champ *speed* possède la même valeur que le paramètre qui suit seront utilisés pour contacter le système distant décrit ici ;
- speed : ce champ est utilisé conjointement avec le champ précédent ;
- phone-number : ce champ indique le numéro de téléphone à composer pour contacter le système distant ;
- chat-script : il s'agit ici du script de *login*. Il est exécuté lorsque la connexion est physiquement établie entre les deux modems, et son but est de permettre l'identification de type *login/password*. Son format est le suivant :

  ```
  expect send expect send ...
  ```

  Un retour chariot est implicite après chaque envoi. Les chaînes `expect` peuvent aussi prendre la forme `expect1-send-expect2`, et dans ce cas, cela indique que si `expect1` n'est pas reconnu au terme d'un certain délai, `send` est renvoyé et `expect2` est attendu en réponse. Ainsi, `in:--in:` est utilisé pour envoyer un retour chariot si on n'a pas reçu `login:`.

Dans le cas qui nous occupe, nous allons utiliser la configuration suivante :

```
193.168.100.1 Any ACU 38400 0,0123456789 in:--in: lucstone word: EliBeurt
```

2. `/etc/ppp/Devices` - ce fichier fournit des informations sur les pilotes de périphérique disponibles. Son format est le suivant :

   ```
   dialer device  speed   [optional]
   ```

   Voici la signification de ces paramètres :

   - dialer : ce paramètre indique le nom d'un *dialer*. Un *dialer* est un script défini dans le fichier `/etc/ppp/Dialers`, qui permet au système de configurer le modem et de lui demander de se connecter à l'équipement distant. Une fois ce script exécuté, c'est le script de *login* défini dans `/etc/ppp/Systems` qui entre en jeu. Un ensemble de *dialers* est déjà défini dans le fichier fourni par défaut.

- device : ce paramètre prend pour valeur le nom d'un pilote de périphérique présent dans le répertoire `/dev` ;
- speed : ce paramètre indique, par une chaîne de caractères alpha-numériques au choix de l'utilisateur, le débit disponible sur le périphérique. La valeur de ce champ doit être la même que dans le fichier `/etc/ppp/Systems` ;
- des paramètres facultatifs peuvent être fournis.

Dans le cas qui nous occupe, nous allons utiliser la configuration suivante :

```
USRcng  cua    38400       rtscts
```

3. `/etc/ppp/Dialers` - ce fichier fournit des informations sur les *dialers*. Son format est le suivant :

```
dialer chat-script
```

Parmi les *dialers* déjà définis, on trouve par exemple celui correspondant au modèle Sportster de la gamme de modems USRobotics :

```
USR-SPORTSTER   ABORT BUSY ABORT ERROR \
        ABORT NO\sCARRIER ABORT NO\sDIAL\sTONE ABORT NO\sANSWER \
        TIMEOUT 5 "" AT OK-ATQ2V1-OK ATD\T TIMEOUT 60 CONNECT
```

4. `/etc/ppp/Filters` - ce fichier indique les filtres qui vont être appliqués à tous les datagrammes qui vont transiter sur la ligne. A chacune des règles de filtrage est associée une action. Cela permet de mettre en place une politique de sécurité sur le site, mais aussi de contrôler les datagrammes susceptibles d'activer et de maintenir la ligne. Nous ne modifierons pas ce fichier dans le cadre de notre exemple, afin de garder un comportement ouvert par défaut ;
5. `/etc/ppp/Auth` - ce fichier contient les informations utilisées pour l'authentification PAP et CHAP. Son format est le suivant :

```
name    secret          [address]
```

Voici la signification de ces paramètres :

- name : il s'agit de l'adresse IP ou du nom du système distant ;
- secret : pour l'authentification PAP, il s'agit du mot de passe, pour l'authentification CHAP il s'agit du secret ;
- address : ce champ facultatif spécifie les adresses IP qui pourront être négociées avec le système distant.

Dans le cas qui nous occupe, nous allons utiliser la configuration suivante, que le fournisseur doit nous avoir indiquée :

```
routeurFournisseur   leSecret
```

Maintenant que nous avons convenablement mis au point les fichiers de configuration, nous allons modifier le script `/etc/ppp/Startup`. Le rôle de ce script est de charger des modules noyau nécessaires au fonctionnement de PPP. Nous devons ajouter, en dernière ligne, la commande d'activation du démon PPP :

```
/usr/etc/pppd auto requirechap name fenetre 192.168.22.33:192.168.100.1
```

L'option `auto` permet le fonctionnement en mode démon ; la phase d'identification CHAP est imposée par l'option `requirechap` ; l'option `name` indique que le paramètre qui suit est le nom local utilisé dans la phase CHAP.

Le paramètre `192.168.22.33:192.168.100.1` indique que l'interface PPP locale doit porter l'adresse `192.168.22.33` et que le système distant est celui décrit dans le fichier `/etc/ppp/Systems` sur la ligne commençant par `192.168.100.1`. Il s'agit du routeur du fournisseur.

Enfin, une route par défaut doit être activée. La commande Unix suivante permet de la mettre en place :

```
<ls@fenetre> su -
Password:
# route add default 192.168.100.3 1
#
```

Pour finaliser l'installation, il nous suffit de modifier les fichiers de démarrage de la machine pour exécuter le script `/etc/ppp/Startup` et mettre en place la route par défaut. On doit alors redémarrer la machine.

La commande `modstat` permet de constater que les modules se sont chargés correctement :

```
<ls@fenetre> modstat
Id  Type  Loadaddr      Size   B-major  C-major  Sysnum   Mod Name
 2  User  ff08e000      1000                              pppframe
 1  Pdrv  ff08b000      3000              59.             tun
```

3.5 Réseau Numérique à Intégration de Services

3.5.1 Principe

Le Réseau Numérique à Intégration de Services RNIS ou ISDN (Integrated Services Digital Network) est l'aboutissement d'un effort de normalisation international des liaisons numériques entre les abonnés et les opérateurs télécom, dans le but de transporter de la voix, mais

aussi toutes sortes de données, graphiques, images, vidéo, etc. Le réseau RNIS proposé par France Télécom s'appelle Numéris.

Avec RNIS, les données sont transportées sur des canaux B ou D. Les canaux de type B sont destinés à transporter les données utilisateur au débit de 64 Kbits/s, tandis que les canaux de type D sont destinés aux protocoles de signalisation, en suivant notamment les trois niveaux les plus bas de la norme OSI, avec un certain nombre d'extensions. Le débit des canaux D dépend du type d'accès.

Les opérateurs fournissent deux types d'accès à RNIS : l'accès de base nommé BRI (Basic Rate Interface), et l'accès primaire nommé PRI (Primary Rate Interface).

L'accès BRI correspond à la fourniture de deux canaux B et d'un canal D par l'opérateur télécom. Le canal D opère à 16 Kbits/s. L'accès BRI, en tenant compte des bits de synchronisation et autres bits d'*overhead*, correspond ainsi à un débit total de 192 Kbits/s.

Les caractéristiques de l'accès PRI dépendent du lieu géographique : en Amérique du Nord et au Japon, l'accès PRI comprend 23 canaux B et un canal D à 64 Kbits/s, ce qui correspond à un débit total de 1,544 Mbits/s, alors qu'en Europe et dans le reste du monde, l'accès PRI comprend 30 canaux B et un canal D à 64 Kbits/s, pour un débit total de 2,048 Mbits/s.

3.5.2 Installation d'abonné

Deux types d'installations sont proposés : l'installation à bus passif et l'installation avec régie d'abonné.

L'installation à bus passif ne nécessite aucun équipement particulier autre que les terminaux RNIS (par exemple postes téléphoniques RNIS, micro-ordinateurs). Elle correspond à la fourniture par l'opérateur télécom d'un accès de type BRI. Jusqu'à cinq équipements peuvent être placés sur le bus, ils se partageront le canal-D ; deux terminaux au plus pourront établir simultanément des communications par les canaux B.

Sur un même bus, les équipements sont accessibles par un même numéro d'appel, et sont différenciés par leur sous-adresse, qui est un numéro supplémentaire représenté traditionnellement après le numéro d'appel, séparé de celui-ci par un caractère astérisque « * ». Les avantages par rapport à une ligne traditionnelle résident dans les services supplémentaires offerts, comme la possibilité de connaître le numéro d'appelant lors d'un appel entrant ou d'obtenir des informations sur le coût de la communication en cours.

L'installation avec régie d'abonné permet d'offrir des services encore plus intéressants. La régie est un autocommutateur qui, du côté de l'opérateur télécom, accueille un ou plusieurs accès BRI et PRI, et du côté de l'abonné propose soit un bus unique, soit plusieurs bus en étoile. Sur chaque bus, deux canaux B et un canal D sont disponibles. Ce type de bus s'appelle S0.

La régie permet aux terminaux RNIS d'un même abonné de communiquer entre eux sans

passer par l'opérateur télécom, c'est-à-dire sans coût de communication. De plus, des services de transfert d'appel d'un poste à l'autre, de mise en attente, et la gestion des appels par un standard sont très souvent proposés. La régie permet aussi d'attribuer des numéros d'appel distincts à des terminaux connectés sur un même bus (SDA, sélection directe à l'arrivée).

La figure 3.14 reprend les différents types d'installations d'abonné.

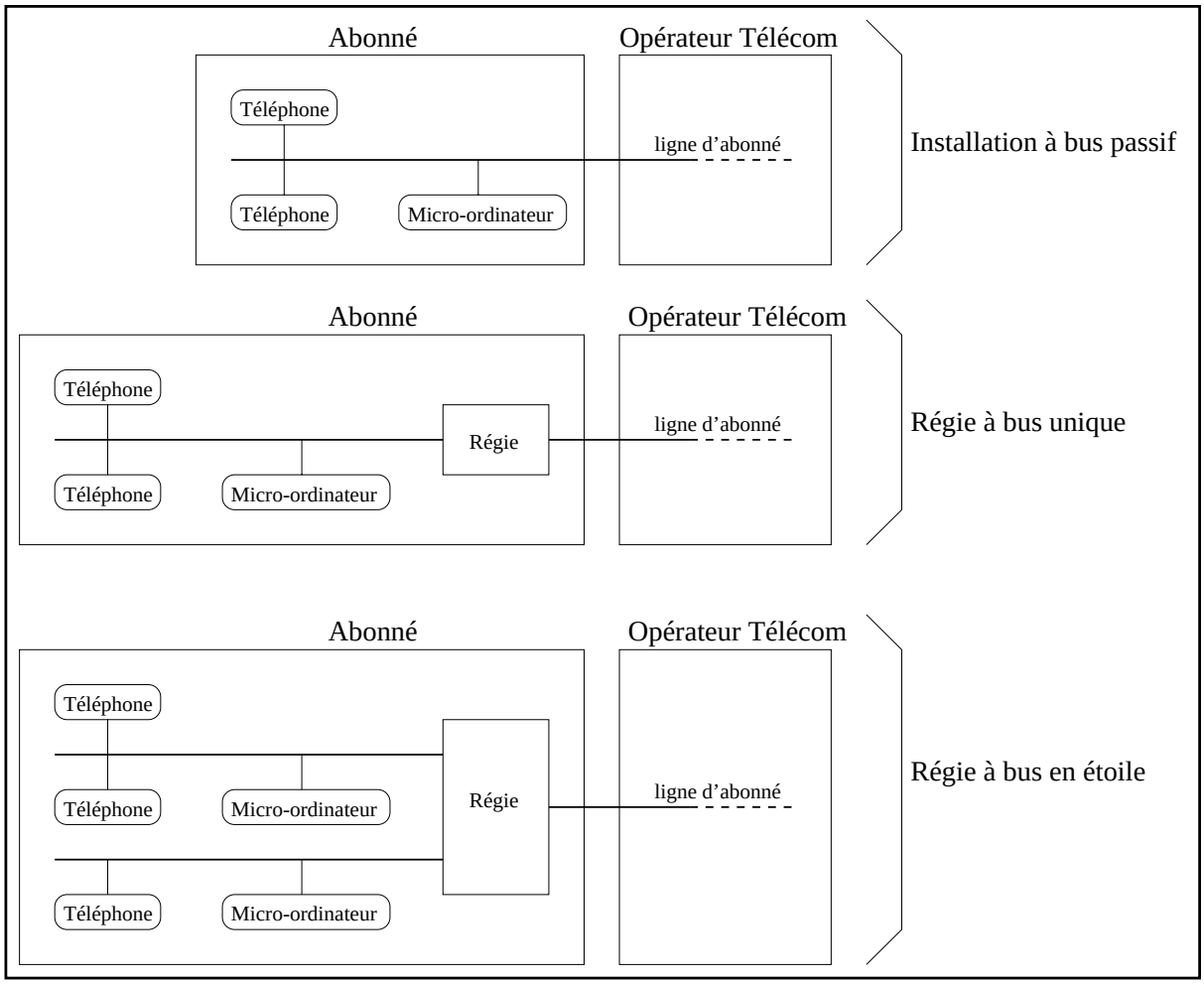

Figure 3.14 *Différents types d'installations RNIS*

On définit deux types d'équipements qui composent la chaîne entre les terminaux de l'abonné et le réseau de communication :

- La Terminaison Numérique de Réseau (TNR ou NT1[8]) appartient à l'opérateur télécom et adapte la ligne d'abonné à la régie d'abonné. Elle fournit notamment des fonctions de maintenance et d'exploitation. L'interface entre la TNR et la ligne d'abonné est appelée interface U, et l'interface entre la régie et la TNR est appelée interface T.
- La Terminaison Numérique d'Abonné (TNA ou NT2[9]) constitue une des composantes

8. Network Termination type 1
9. Network Termination type 2

de la régie d'abonné. Elle fournit un ou plusieurs bus de type S à partir de son interface avec un ou plusieurs Terminaux Numériques de Réseau.

Les terminaux pouvant être raccordés au réseau RNIS sont classés en deux types :

– Les TE1 constituent les terminaux numériques pouvant être reliés directement à un bus S, tel que des postes téléphoniques RNIS, ou des micro-ordinateurs équipés d'une carte d'interface adéquate.

– Les TE2 constituent les terminaux traditionnels ; il peut par exemple s'agir d'un micro-ordinateur équipé d'une interface série. On les raccorde au bus S par l'intermédiaire d'un adaptateur de terminal (TA[10]) qui propose une interface de type R avec le terminal traditionnel et une interface de type S avec le bus.

Les cinq types d'équipements (TE1, TE2, TA, TNA et TNR) ainsi que les quatre types d'interfaces (R, S, T et U) sont schématisés sur la figure 3.15.

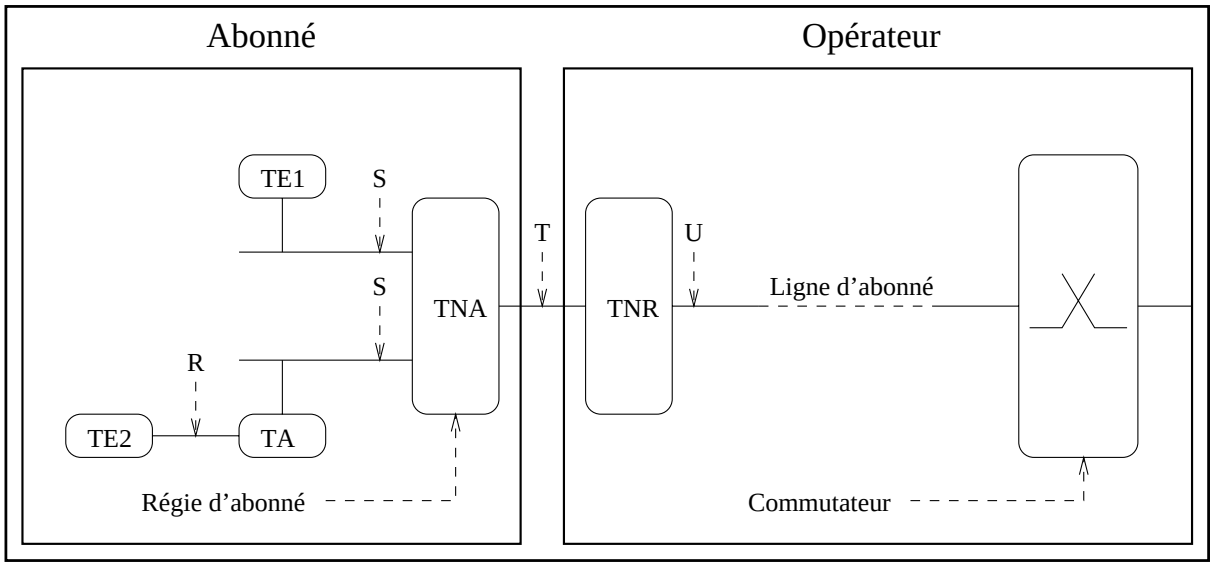

Figure 3.15 *Interfaces et équipements RNIS*

En plus des bus de type S0, on trouve des bus S1 et S2 à des débits plus importants, en fonction du lieu géographique :

Interface	Canaux	Débit utile	Débit total	Zone
S0	2B+D (2*64+16)	144 Kb/s	144 Kb/s	Tous pays
S1	23B+D (23*64+64)	1536 Kb/s	1536 Kb/s	Amérique, Japon
S2	30B+D (30*64+64)	1984 Kb/s	1984 Kb/s	reste du monde

De même que pour les bus de type S, les interfaces T sont disponibles à différents débits,

10. Terminal Adapter

notamment en fonction du lieu géographique :

Interface	Canaux	Débit utile	Débit total	Zone
T0	S0 + 48 Kb/s	144 Kb/s	192 Kb/s	Tous pays
T1	S1 + 8 Kb/s	1536 Kb/s	1544 Kb/s	Amérique, Japon
T2	S2 + 64 Kb/s	1984 Kb/s	2048 Kb/s	reste du monde

3.5.3 Types de services

Les opérateurs font souvent des choix particuliers d'implémentation, et il faut donc indiquer aux terminaux RNIS le type de commutateur chez l'opérateur. Le tableau suivant récapitule les différents types d'opérateurs. En France, les types utilisés sont vn2, vn3 et vn4. Cette information peut être demandée dans la phase de configuration du terminal.

Type	Opérateur
vn2	France
vn3	
vn4	
basic-1tr6	Allemagne
basic-nwnet3	Norvège
basic-net3	Royaume-Uni et autres
primary-net5	Europe et Royaume-Uni (accès primaire)
basic-ts013	Australie
basic-nznet3	Nouvelle Zelande
ntt	Japon (accès de base)
primary-ntt	Japon (accès primaire)
basic-5ess	Amérique du Nord
basic-dms100	
basic-ni1	
primary-4ess	(accès primaire)
primary-5ess	(accès primaire)
primary-dms100	(accès primaire)

Numéris offre de nombreux services en mode commutation de paquets sur canal B ou D. De plus, Numéris offre deux services en mode commutation de circuit sur canal B :

- le service Circuit Commuté sur canal B Transparent (CCBT), appelé service transparent ;
- le service Circuit Commuté sur canal B Non Transparent (CCBNT), appelé service audio.

Il est souvent nécessaire d'indiquer à la régie de passer en mode transparent sur une interface où on relie une passerelle Internet par RNIS. L'oubli de cette intervention peut être une cause de non-fonctionnement de l'accès Internet.

3.5.4 Accès Internet avec adaptateur de terminal

L'accès à l'Internet avec adaptateur de terminal permet de se connecter à l'aide d'un routeur qui ne dispose pas de fonctionnalité matérielle adaptée à RNIS, mais tout simplement d'une interface série et d'une carte d'interface le rattachant au réseau local. On va donc se contenter d'adjoindre un adaptateur de terminal RNIS au routeur pour le transformer en routeur RNIS.

L'adaptateur de terminal est un équipement relié d'un côté au bus S et de l'autre, par l'interface de type R, à un équipement connecté au réseau local. L'adaptateur joue ainsi le rôle d'ETCD, et l'autre équipement d'ETTD. À la différence d'un modem, l'adaptateur de terminal ne transmet pas les informations par modulation, mais directement en bande de base. Ainsi, toute la chaîne de communication entre le réseau local du client et celui de son fournisseur Internet est entièrement numérique.

L'interface de type R est une jonction V24/V28, V24/V35, X24/V11 ou analogique. Dans ce dernier cas, on peut relier par exemple un poste téléphonique classique. Dans le cadre de la connexion avec un fournisseur Internet, on choisit le plus souvent un adaptateur de terminal muni d'une prise V24/V28.

La jonction qui relie l'adaptateur à l'ETTD peut être synchrone ou asynchrone. Dans le cas d'une jonction synchrone, on aura à disposition un débit de 64 Kbits/s appelé « mode d'adaptation transparent », correspondant à un canal B, et dans le cas d'une jonction asynchrone on aura un débit pouvant atteindre 57 600 bits/s. Un adaptateur de terminal peut fournir des débits plus faibles sur sa jonction en mode asynchrone, mais c'est toujours un canal B complet qui est utilisé. Lorsque l'interface V24/V28 de l'ETTD n'autorise par exemple pas plus de 19 200 bits/s, on peut dégrader le fonctionnement de l'adaptateur de terminal pour qu'il utilise ce débit, mais c'est une solution à éviter car il est dans ce cas plus économique d'utiliser un accès par modem au même débit. De plus, le fournisseur doit lui aussi posséder un équipement configuré à 19 200 bits/s. Tous n'offrent pas ce service dégradé, qui leur demande l'acquisition d'un matériel dédié au service.

Adaptateur de terminal en mode asynchrone

L'utilisation la plus répandue d'un adaptateur de terminal consiste à se connecter à un fournisseur en mode asynchrone à 57 600 bits/s. En effet, c'est le plus grand débit possible en mode asynchrone. Les connexions à des débits plus importants imposent d'utiliser un micro-ordinateur ou une station de travail disposant d'une interface série synchrone, ce qui nécessite le plus souvent l'achat d'une carte d'extension série synchrone et d'une passerelle PPP adaptée en plus de l'adaptateur de terminal, le coût de l'opération approchant le prix d'un routeur RNIS dédié intégrant la fonction PPP et ne nécessitant pas l'utilisation d'un adaptateur de terminal, comme nous le verrons dans la section 3.5.5 page 108.

Un adaptateur de terminal en mode asynchrone permet d'adapter un flux de caractères d'une jonction asynchrone à un canal B, canal de transmission numérique au niveau bit. Il utilise

pour cela un protocole particulier, tel que défini dans les avis V110 et V14 étendu de l'ITU-T, afin de repérer les différents caractères dans le flux de bits, et d'adapter la vitesse de jonction avec le débit de 64 Kbits/s du canal B.

Il faut donc configurer l'adaptateur de terminal en lui précisant le débit et le protocole utilisé. On utilise pour cela des commandes de type AT, suivant le même principe que la configuration d'un modem. Par exemple, l'adaptateur TELSAT-3202S de la SAT[11] se configure à l'aide des commandes suivantes :

Débit de jonction	Protocole d'adaptation	Commande AT
Configuration automatique	V110	%A0%F0
600 bits/s	V110	%A0%F3
1 200 bits/s	V110	%A0%F4
2 400 bits/s	V110	%A0%F5
4 800 bits/s	V110	%A0%F6
9 600 bits/s	V110	%A0%F7
19 200 bits/s	V110	%A0%F8
38 400 bits/s	V110	%A0%F9
57 600 bits/s	V14 étendu	%A0%F12
64 000 bits/s	mode transparent	C5

Pour composer un numéro d'appel, on utilise, comme pour un modem, la commande `D` : `ATDn*s` où `n` représente le numéro d'appel et `s` la sous-adresse. S'il n'y a pas de sous-adresse, on utilise simplement `ATDn`.

Sachant qu'un adaptateur de terminal peut fournir une synchronisation au niveau caractère, tout type de protocole prévu pour des jonctions asynchrones peut être utilisé pour la connexion RNIS par ce biais. Notamment, on peut utiliser UUCP ou SLIP décrits dans les sections 3.4.3 page 87 et 3.4.4 page 88. Mais les fournisseurs Internet proposent pour la plupart uniquement des services PPP de ce type, car PPP fournit plus de services qu'UUCP et permet un meilleur contrôle de la ligne que SLIP.

La figure 3.16 page suivante présente les différents équipements qui entrent en jeu lors d'une connexion PPP avec un adaptateur de terminal :

- Chez le client, on trouve un équipement muni de l'adaptateur de terminal, et branché sur un brin quelconque du réseau local, par une carte Ethernet dans notre exemple ; L'adaptateur de terminal est relié à un autocommutateur qui intègre la fonction TNA, il est tout à fait possible de s'en passer si on prend une installation à bus passif, comme nous l'avons vu précédemment.
- Le fournisseur possède quant à lui un autocommutateur relié à une TNR. Cet autocommutateur distribue les appels sur les TA libres. Les TA sont reliés à un concentrateur de terminaux par des jonctions V24/V28 asynchrones. Le concentrateur est relié au

11. La SAT appartient au groupe SAGEM

réseau local du fournisseur, par exemple par une interface Ethernet. Il joue alors le rôle de serveur PPP et c'est lui qui va identifier les clients lors de leurs appels, négocier les paramètres de la connexion et permettre le transport des datagrammes IP sur la ligne série. Il faut noter que certains fournisseurs s'équipent de matériel qui assure tout à la fois la fonction autocommutateur, adaptateurs de terminaux, concentrateur de terminaux et serveur PPP.

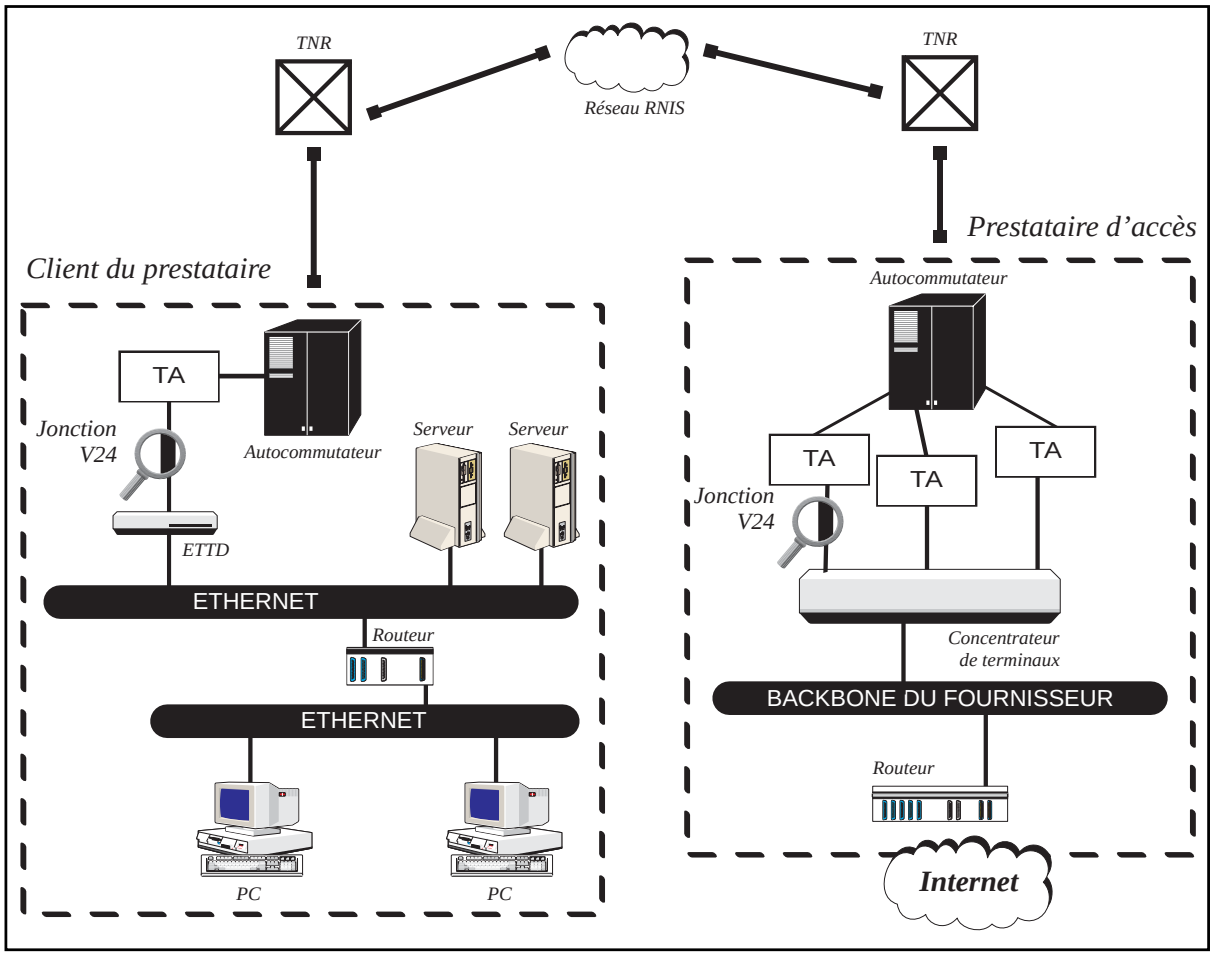

Figure 3.16 *Connexion RNIS avec adaptateur de terminal : topologie*

On peut faire un parallèle simple avec la connexion PPP par modem abordée dans la section 3.4.5. Au niveau de la définition du plan d'adressage IP, il suffit de remarquer qu'ici, l'adaptateur de terminal remplace le modem, et que le plan d'adressage est ainsi identique car on utilise un même protocole : PPP. La figure 3.17 page 109 présente donc ce plan d'adressage IP. Comme lors du raccordement avec modem, on utilise ici souvent une interface PPP de type point à point. De même, la mise en place du routage est exactement la même que lors de la connexion avec modem. De plus, le logiciel PPP utilisant simplement une interface série sur le micro-ordinateur ou la station de travail, toutes les passerelles PPP logicielles pour Macintosh, PC ou Unix présentées précédemment dans le cadre d'un raccordement par modem peuvent être utilisées, et elles apporteront les mêmes fonctionnalités. Le lecteur est donc prié

de s'y référer pour le détail de cette mise en place.

Adaptateur de terminal en mode synchrone transparent

Comme nous l'avons dit précédemment, ce mode de connexion permet d'obtenir un débit de 64 Kbits/s. Il nécessite l'acquisition de trois modules :

- un adaptateur de terminal fournissant le mode d'adaptation dit « transparent » ;
- une carte série synchrone sur un micro-ordinateur ou station de travail ;
- une passerelle PPP logicielle permettant d'utiliser la carte série et disposant du protocole PPP sur jonction synchrone.

Pour configurer l'adaptateur de terminal, il faut le faire passer en mode synchrone, s'il ne l'est déjà. Pour ce faire, une commande de type AT est fournie. Avec l'adaptateur TELSAT-3202S, on utilise la commande `C5`.

Configuration d'un TA synchrone : commandes V25 bis

Une fois en mode synchrone, on ne peut évidemment plus utiliser un émulateur de terminal pour envoyer des commandes AT à l'adaptateur : rappelons qu'en mode synchrone, la notion de caractère disparaît. Il faut donc utiliser le protocole défini par l'avis V25bis de l'ITU-T, qui permet de communiquer des commandes sur une jonction synchrone. On n'utilise donc plus un émulateur de terminal mais directement la passerelle PPP synchrone pour envoyer les commandes V25bis sur la jonction. Par exemple, pour demander à l'adaptateur TELSAT-3202S de composer un numéro d'appel, on utilise la commande V25bis `CRNn*s` où `n` représente le numéro d'appel et `s` représente la sous-adresse. S'il n'y a pas de sous-adresse, on utilise simplement la commande `CRNn`.
Certains équipements disposent d'une jonction asynchrone en supplément de leur jonction synchrone, prévue uniquement pour la phase de configuration et de commande ; on parle alors de configuration hors-bande.

Au niveau plan d'adressage et configuration IP, le principe est le même qu'avec un adaptateur de terminal en mode asynchrone.

Notons qu'il est souvent plus simple d'acquérir un routeur RNIS spécialisé, solution abordée dans la section 3.5.5.

3.5.5 Accès Internet RNIS avec routeur spécialisé

Protocoles et débits

Lorsqu'on veut accéder à Internet par RNIS à 64 Kbits/s, on fait souvent l'acquisition d'un routeur spécialisé, ou d'une carte RNIS qu'on rajoute à un PC ou à une station de travail. Sur un canal B, en mode transparent, deux principales normes d'encapsulation des datagrammes IP sont disponibles : IP sur X25 sur RNIS, et IP sur PPP synchrone sur RNIS. Ces

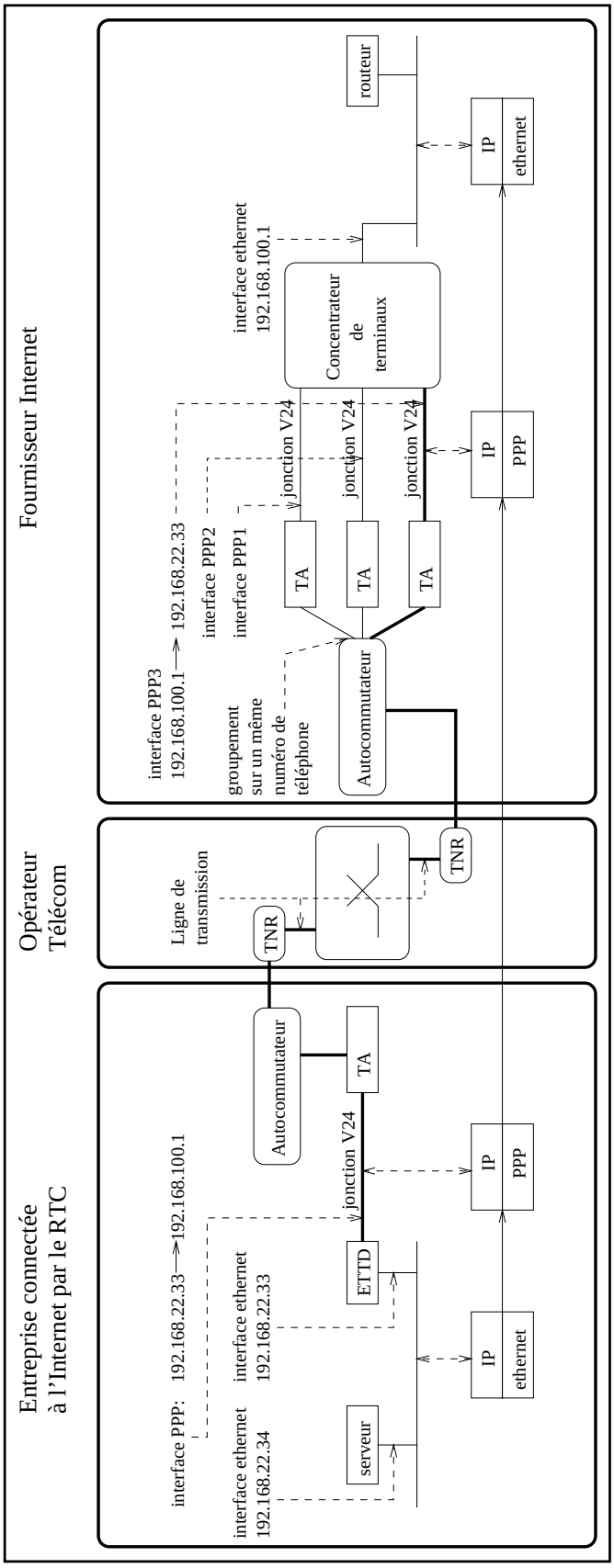

Figure 3.17 *Connexion RNIS avec adaptateur de terminal : plan d'adressage*

deux modes de connexions sont récapitulés avec un exemple d'acheminement d'informations World Wide Web d'une transaction HTTP, sur la figure 3.18 pour la connexion IP sur X25 et sur la figure 3.19 pour la connexion IP sur PPP.

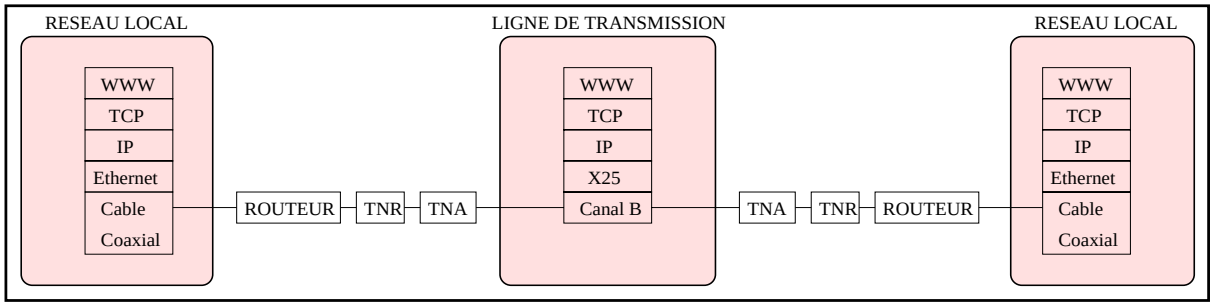

Figure 3.18 *Connexion IP sur X25 sur RNIS*

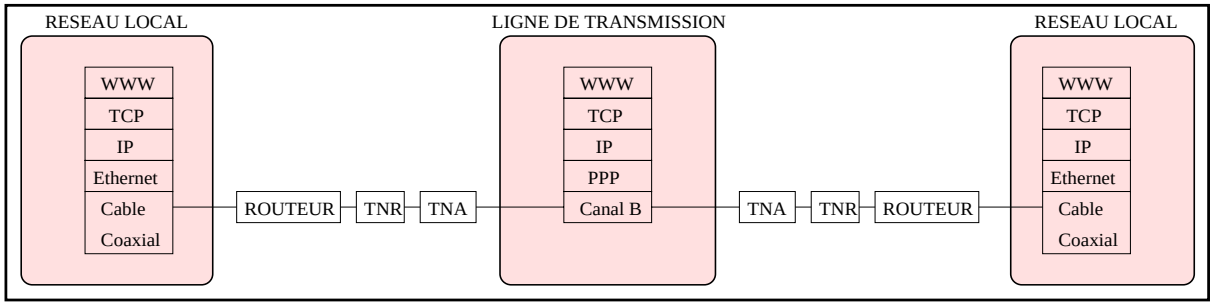

Figure 3.19 *Connexion IP sur PPP sur RNIS*

Un canal B est utilisé avec la connexion IP sur RNIS, qui permet donc un débit de 64 Kbits/s. Certains routeurs RNIS permettent d'utiliser deux canaux B en IP sur PPP sur RNIS, et ainsi d'obtenir un accès à 128 Kbits/s, mais les techniques mises en œuvre pour utiliser conjointement deux canaux B n'ont été normalisées complètement que récemment, et cela a empêché souvent ce type de routeur d'interopérer avec un équipement d'une autre marque à ce débit. C'est pourquoi nombre de fournisseurs proposent un débit sur RNIS limité à un unique canal B, ou imposent la marque du routeur RNIS que le client doit acquérir.

Néanmoins, une norme permettant l'utilisation de plusieurs canaux de données a été proposée sous la forme de la publication d'un RFC (RFC 1717), afin de permettre à deux passerelles PPP d'utiliser un nombre quelconque de lignes de transmission pour échanger des datagrammes. Le développement de cette fonctionnalité, appelée PPP MP (PPP Multilink Protocol), a été motivé par le besoin d'utiliser plusieurs canaux B sur une même connexion RNIS. Actuellement, peu de fournisseurs proposent de tels accès, mais de plus en plus de passerelles PPP disposent de cette fonctionnalité. Notons qu'il existe un autre type de couplage, au niveau RNIS cette fois-ci, qui permet de fusionner des canaux B indépendamment des protocoles utilisés au-dessus. Cette technique est appelée *bonding* (Bandwidth on Demand Interoperability Group).

Nous allons nous attacher maintenant à décrire l'accès Internet par RNIS le plus courant :

64 Kbits/s sur un canal B, avec IP sur PPP synchrone. L'utilisation avec plusieurs canaux B consiste à ajouter des commandes à la configuration de l'équipement RNIS afin de lui fournir les numéros d'appel correspondant aux différents canaux, dans le cas où ces numéros sont distincts. Tout ce qui suit est donc applicable autant à 64 Kbits/s qu'à des multiples de ce débit.

Équipements

La figure 3.20 page suivante présente les différents équipements qui entrent en jeu lors d'une connexion RNIS avec un routeur spécialisé :

- Chez le client, on trouve un routeur RNIS connecté par une interface Ethernet au réseau local, et muni d'une interface BRI connectée au bus S0 d'un autocommutateur numérique. Cet équipement fait office de passerelle PPP, et fournit souvent d'autres services tels que la journalisation[12] des datagrammes qui le traversent ou le filtrage des données qui le traversent pour sécuriser l'accès au site.
- Le fournisseur possède quant à lui le plus souvent un routeur directement relié à une TNR, permettant de gérer avec un seul équipement jusqu'à 30 canaux B. Ce routeur est muni d'une interface Ethernet et joue ainsi le rôle de serveur PPP. C'est lui qui va identifier les clients lors de leurs appels, par exemple avec PAP ou CHAP, négocier les paramètres de la connexion, permettre le transport des datagrammes IP entre la TNR et le réseau local, et enregistrer les informations de facturation : temps de connexion, volume de transfert effectué. La base de données d'informations sur les utilisateurs, contenant les adresses IP des machines distantes, leur mot de passe PAP ou les secrets CHAP, n'est pas toujours maintenue sur le routeur RNIS du fournisseur : ce routeur fait parfois appel à un serveur Unix connecté au réseau local du fournisseur afin d'obtenir ces informations lors des connexions des clients. Il utilise alors un des trois protocoles suivants : RADIUS, TACACS ou TACACS+.

Plan d'adressage

Nous avons vu section 3.4.5 page 89 qu'il existe deux types d'interfaces, en ce qui concerne le plan d'adressage : les interfaces multipoints et les interfaces point à point. Jusqu'à maintenant, nous avons étudié le plan d'adressage de connexions PPP sur paire de modems, PPP sur adaptateur de terminal RNIS asynchrone et PPP sur adaptateur de terminal RNIS en mode transparent. Les trois plans d'adressage associés utilisaient tous, autant du côté du client que du fournisseur Internet, une interface LAN multipoint[13] et une interface PPP point à point.

Un routeur spécialisé possède de même une interface LAN multipoint. Le fournisseur Internet demande souvent à son client de configurer son interface PPP en multipoint, et lui fournit

12. *accounting*
13. Le plus souvent une interface Ethernet

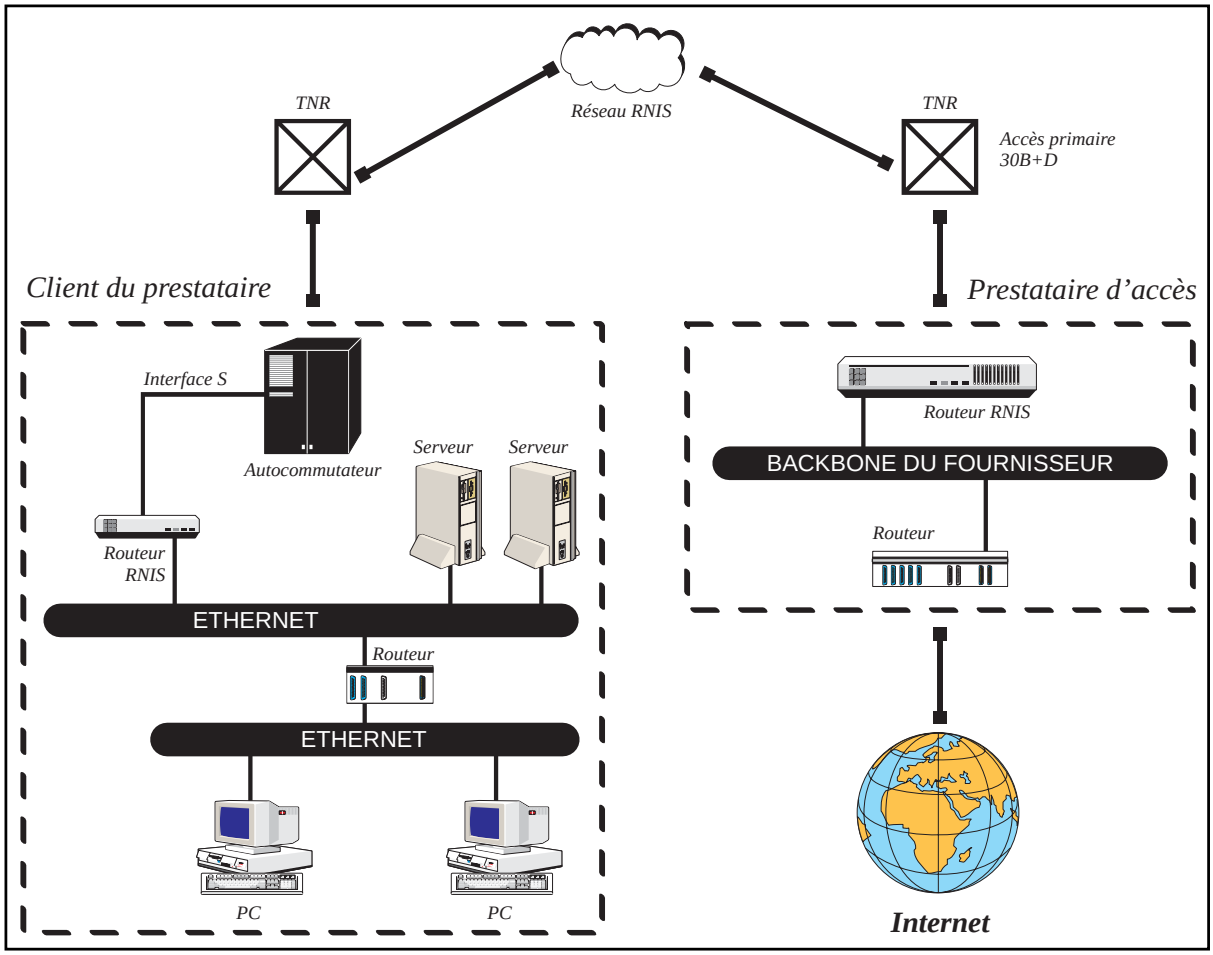

Figure 3.20 *Connexion RNIS avec routeur dédié : topologie*

pour cela un réseau particulier appelé *réseau d'interconnexion*. Notons bien qu'il est tout à fait possible d'utiliser une interface PPP point à point avec un routeur spécialisé, mais que c'est rarement l'option choisie. Si le cas se présentait, le lecteur serait tout simplement amené à se référer à la description du plan d'adressage décrit section 3.4.5 page 89.

Le réseau d'interconnexion attribué par le fournisseur correspond à une classe C entière, ou parfois à un sous-réseau d'une classe C. Ce réseau va accueillir les adresses IP des interfaces PPP du fournisseur et de ses clients. Nous verrons dans la section 3.6 page 122 un autre type de réseau d'interconnexion : le sous-réseau d'interconnexion n'accueillant que le fournisseur et un seul de ses clients.

Reprenons l'exemple de réseau client utilisé au chapitre 2 pour présenter les masques de sous-réseaux.

Rappelons que ce réseau possède un masque de sous-réseaux de valeur `255.255.255.224`, et trois brins Ethernet répartis sur les trois premiers sous-réseaux utilisables dans le réseau de classe C `192.168.22.0`. Le routeur connecté à l'Internet possède une interface Ethernet d'adresse `192.168.22.33` sur le réseau local.

Considérons aussi que le fournisseur possède trois autres clients ayant souscrit au service d'accès RNIS 64 Kbits/s, et qu'il leur a attribué les classes C suivantes : `192.168.23.0`, `192.168.24.0` et `192.168.25.0`. Le client possédant le réseau `192.168.22.0` sera désigné par client A, ces trois autres clients par B, C et D. Le fournisseur possède quant à lui un réseau local constitué par la classe C `192.168.100.0`. Nous pouvons donc récapituler les réseaux de classe C en jeu dans le tableau suivant :

Entité	**Adresse réseau**	**masque de sous-réseaux**
Client A	`192.168.22.0`	`255.255.255.224`
Client B	`192.168.23.0`	N.C.
Client C	`192.168.24.0`	N.C.
Client D	`192.168.25.0`	N.C.
Fournisseur	`192.168.100.0`	`255.255.255.0` (sans découpage en sous-réseaux)
Réseau d'interconnexion	`192.168.200.0`	`255.255.255.0` (sans découpage en sous-réseaux)

Les adresses IP des différentes interfaces des équipements en jeu lors des connexions des clients sont les suivantes :

Entité	**Interface**	**Adresse IP**	**Réseau**	**Masque de sous-réseaux**	**Adresse de diffusion**
Client A	Ethernet	`192.168.22.33`	`192.168.22.0`	`255.255.255.224`	`192.168.22.63`
	PPP	**`192.168.200.2`**	`192.168.200.0`	`255.255.255.0`	`192.168.200.255`
Client B	Ethernet	`192.168.23.50`	`192.168.23.0`	N.C.	N.C.
	PPP	**`192.168.200.3`**	`192.168.200.0`	`255.255.255.0`	`192.168.200.255`
Client C	Ethernet	`192.168.24.91`	`192.168.24.0`	N.C.	N.C.
	PPP	**`192.168.200.4`**	`192.168.200.0`	`255.255.255.0`	`192.168.200.255`
Client D	Ethernet	`192.168.25.72`	`192.168.25.0`	N.C.	N.C.
	PPP	**`192.168.200.5`**	`192.168.200.0`	`255.255.255.0`	`192.168.200.255`
Fournisseur	Ethernet	`192.168.100.1`	`192.168.100.0`	`255.255.255.0`	`192.168.100.255`
	PPP	**`192.168.200.1`**	`192.168.200.0`	`255.255.255.0`	`192.168.200.255`

Notons que, comme on l'a dit précédemment, toutes les adresses IP des interfaces PPP des clients et du fournisseur se retrouvent dans le même réseau de classe C, dit réseau d'interconnexion. De plus, le routeur du fournisseur possède une seule interface PPP pour tous les clients qui peuvent se connecter, alors que dans le cadre des accès avec modem ou adaptateur de terminal, le fournisseur possède une interface PPP par client connecté à un instant donné. C'est tout à fait compréhensible si on remarque que lors des connexions par modem ou par adaptateur de terminal, le fournisseur possède, sur son routeur, une jonction physique dédié par adaptateur ou par modem, alors qu'ici, une seule interface physique est utilisée pour l'accès au réseau RNIS, en l'occurrence une interface de type T2. Cela explique pourquoi on choisit, dans le cas d'un raccordement par routeur dédié, d'utiliser des interfaces multipoints appartenant à un même réseau d'interconnexion, et pourquoi, dans le cas d'un raccordement par adaptateur de terminal ou par modem, on choisit des interfaces point à point sans réseau d'interconnexion.

La figure 3.21 page ci-contre indique les différentes interfaces en jeu dans le cadre du raccordement du client A, ainsi que leurs adresses IP.

Mise en place du routage

De même qu'avec les autres types de raccordement, le fournisseur se charge d'annoncer le réseau de classe B ou C du client, sur l'Internet par le protocole BGP-4. Les paquets à destination de ce réseau sont ainsi attirés vers le réseau du fournisseur. Ce dernier met de plus en place un routage interne afin que les paquets qui arrivent chez lui soient correctement acheminés vers son client.

Le client doit, quant à lui, mettre en place un routage permettant aux paquets à destination de l'Internet de quitter son réseau local à travers l'interface PPP et d'entrer ainsi chez le fournisseur. Pour cela, la passerelle PPP doit posséder une route par défaut vers son homologue chez le fournisseur dans le réseau d'interconnexion, c'est-à-dire vers `192.168.200.1`. Les autres équipements du réseau local du client doivent posséder une route par défaut vers l'interface Ethernet du routeur RNIS, c'est-à-dire vers `192.168.22.33` pour le client A. Ces routes par défaut peuvent être mises en place en installant un protocole de routage interne, ou en imposant des routes statiques dans les configurations des équipements.

Un problème nouveau peut se poser quand deux clients du service RNIS du même fournisseur veulent dialoguer. Plus précisément, si une machine quelconque du client A essaye de dialoguer avec le routeur RNIS de B, par exemple par l'outil `telnet` qui permet d'accéder à un nœud distant, des problèmes peuvent apparaître. Le client A va pour cela spécifier le nom du routeur de B à son application `telnet`. Ce routeur possède deux interfaces, donc deux adresses IP. Il a donc deux noms, par exemple :

- `routeur-ppp.clientB.fr`
- `routeur-ether.clientB.fr`

Si le client A spécifie le nom `routeur-ppp.clientB.fr` qui correspond à l'adresse de l'interface PPP du routeur de B, l'application `telnet` ne pourra pas communiquer correctement.

En effet, les paquets IP émis par A à destination de B porteront une adresse IP destination de valeur `192.168.200.3`, c'est-à-dire l'adresse du routeur de B dans le réseau d'interconnexion.

Quand ces paquets vont entrer dans le routeur de A, ce dernier va tenter de déterminer le prochain équipement auquel il doit les retransmettre. Mais il ne suivra pas la route par défaut vers le fournisseur car l'adresse de `192.168.200.3` est justement dans le même réseau que celui de son interface PPP. Rappelons qu'une des caractéristiques de l'algorithme de routage IP décrit section 2.6.1 page 49 consiste à ne faire appel à la route par défaut que lorsque l'adresse IP destination n'appartient pas à un réseau directement connecté.

Pour pallier ce problème, si un client A désire dialoguer avec le routeur d'interconnexion

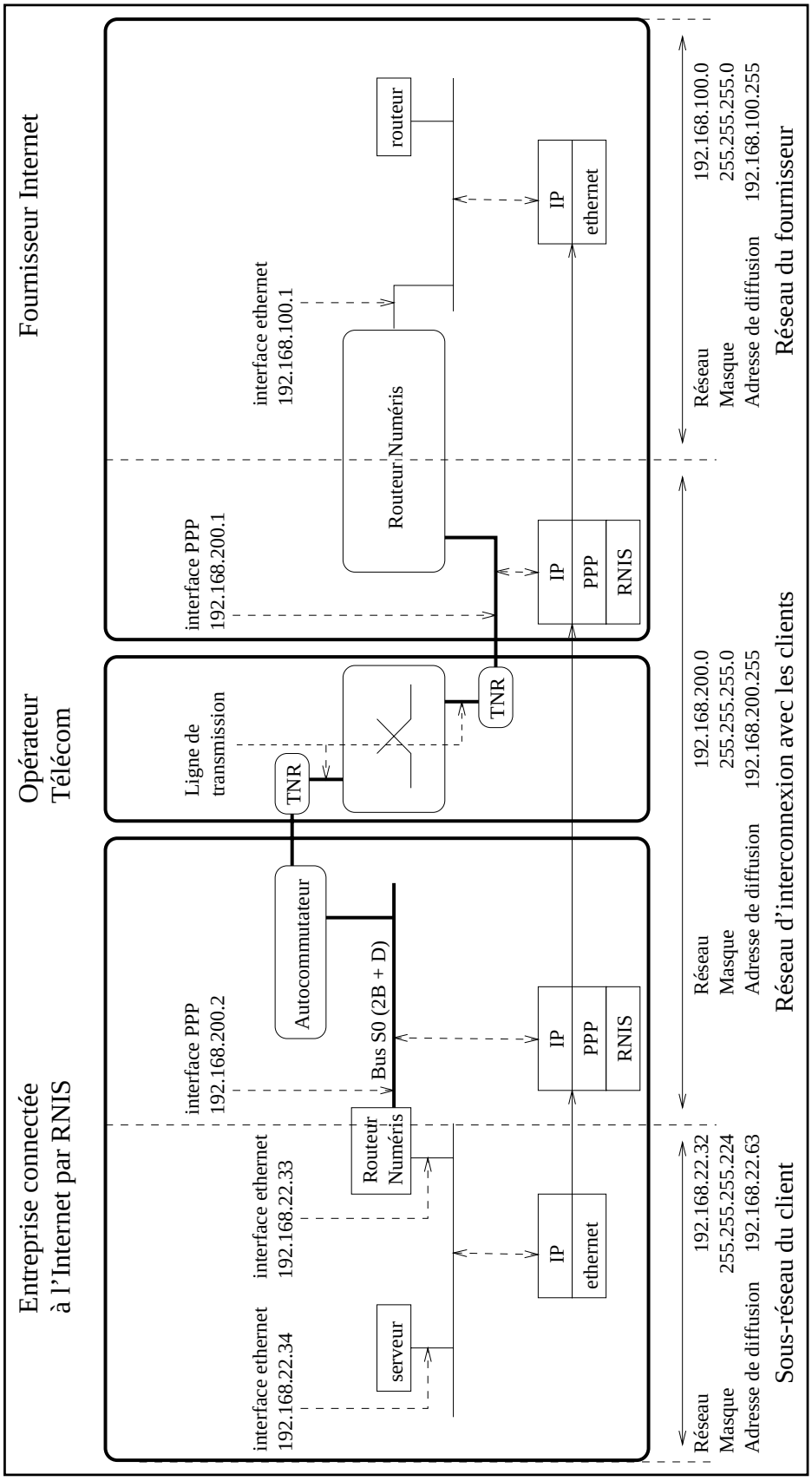

Figure 3.21 *Connexion RNIS avec routeur dédié : plan d'adressage*

de B, il suffit à A d'ajouter sur son routeur une route particulière dite « route de host », qui force le routeur à acheminer un datagramme vers l'interface PPP du fournisseur, d'adresse `192.168.200.1`, lorsque l'adresse destination de ce datagramme est `192.168.200.3`, c'est-à-dire l'adresse de l'interface PPP du routeur de B. Pour les mêmes raisons, si A désire accéder depuis son propre routeur RNIS à des équipements de B, il doit demander à ce dernier de rajouter une « route de host » du même type, sinon les datagrammes risquent de bien parcourir le réseau de A vers B, mais de ne pas revenir.

Routage sur un réseau d'interconnexion partagé entre les clients

On peut généraliser cette méthode par la remarque suivante : quand l'interface WAN du routeur du client appartient à un réseau d'interconnexion partagé avec les autres clients, il faut rajouter sur le routeur local une route à destination du fournisseur, pour ce réseau d'interconnexion.

Ce problème apparaît plus souvent qu'on ne pourrait le croire, car on transforme parfois une station de travail en routeur RNIS par l'adjonction d'une carte RNIS spécialisée, munie d'un accès BRI. On est alors souvent tenté de mettre des services sur cette station, par exemple un serveur WWW ou FTP, qui sera accessible depuis l'extérieur, notamment par les autres machines du réseau d'interconnexion.

3.5.6 Particularités de la connexion RNIS

L'exploitation de la connexion RNIS possède des caractéristiques analogues au raccordement intermittent par modem abordé section 3.4.5 page 92 :

- un client connecté par ce biais ne peut pas héberger de serveur ;
- lorsque le client n'est pas connecté, les articles de News et les *mails* sont placés en attente chez le fournisseur ; le client doit donc se connecter régulièrement pour permettre aux serveurs du fournisseur de faire parvenir les articles et le courrier s'il y en a en attente.

La différence notable entre l'accès RNIS et l'accès par modem se situe dans le temps de mise en place de la connexion : il faut moins d'une seconde à un routeur RNIS pour appeler le routeur du fournisseur et établir les couches de protocoles permettant d'interopérer, alors que dans le cas d'un accès modem, il faut parfois plus de 30 secondes. Ainsi, grâce au Dial-on-Demand, la notion d'accès intermittent disparaît pour l'utilisateur.

Notons de plus qu'une fois configuré, un accès RNIS par routeur spécialisé reste stable, alors qu'un raccordement par modem nécessite parfois des interventions en phase d'exploitation. Par exemple, la fonctionnalité dite des « numéros brûlés », qui interdit à un modem de composer plus d'un certain nombre de fois un même numéro sans succès, impose d'intervenir sur le modem si les lignes du fournisseur sont toutes occupées pendant plusieurs appels successifs. Cette mésaventure survient avec certains fournisseurs qui possèdent très peu de modems par rapport au nombre de clients en accès intermittent dont ils assurent le raccordement.

3.5.7 Quelques routeurs RNIS spécialisés

Parmi les routeurs RNIS les plus utilisés, notons les routeurs des sociétés CISCO et ASCEND.

On peut aussi transformer une station de travail ou un micro-ordinateur en routeur RNIS par l'adjonction d'une carte RNIS spécialisée, ainsi qu'un logiciel adéquat.

On peut pour cela par exemple citer le produit SUNLink-ISDN de SUN Microsystems. Il permet de transformer une station de travail Unix Sparc sous le système d'exploitation Solaris en l'équipant d'une carte munie d'une interface S0 et d'un passerelle PPP logicielle permettant notamment le Dial-on-Demand.

Mais il n'est pas rare qu'une station ainsi équipée atteigne un coût bien supérieur à celui d'un routeur RNIS d'entrée de gamme.

Routeurs CISCO 1003 et 1004

Le CISCO 1003 est particulièrement adapté à la connexion par RNIS : il dispose d'une interface Ethernet 10BaseT (RJ-45), d'un port console (RJ-45) et d'un accès BRI (RJ-45) correspondant à une interface S ou T.

En Amérique du Nord, la TNR n'est pas fournie par l'opérateur mais par son client : c'est donc une interface U qui est nécessaire sur le routeur. On utilise pour cela un CISCO 1004 qui est ainsi destiné aux USA, alors que le CISCO 1003 sera utilisé par exemple en France.

Les figures 3.22 et 3.23 page suivante montrent les faces avant et arrière du modèle 1003, ainsi que ses différents connecteurs.

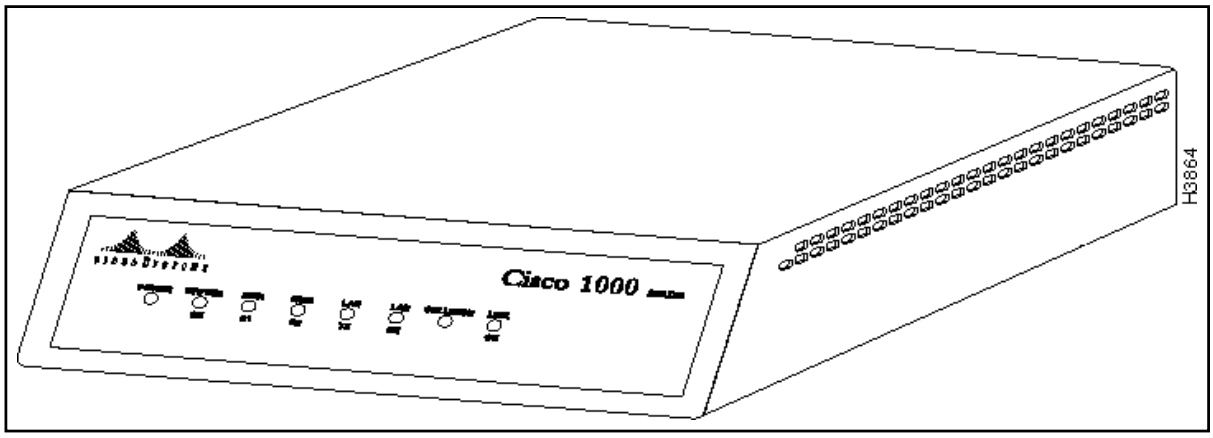

Figure 3.22 *CISCO 1003 - face avant (publié avec l'aimable autorisation de CISCO)*

Examinons les particularités de la configuration d'un routeur CISCO disposant d'un port BRI. Le lecteur est prié de se reporter section 3.3 page 73 pour une introduction à la configuration d'un routeur CISCO.

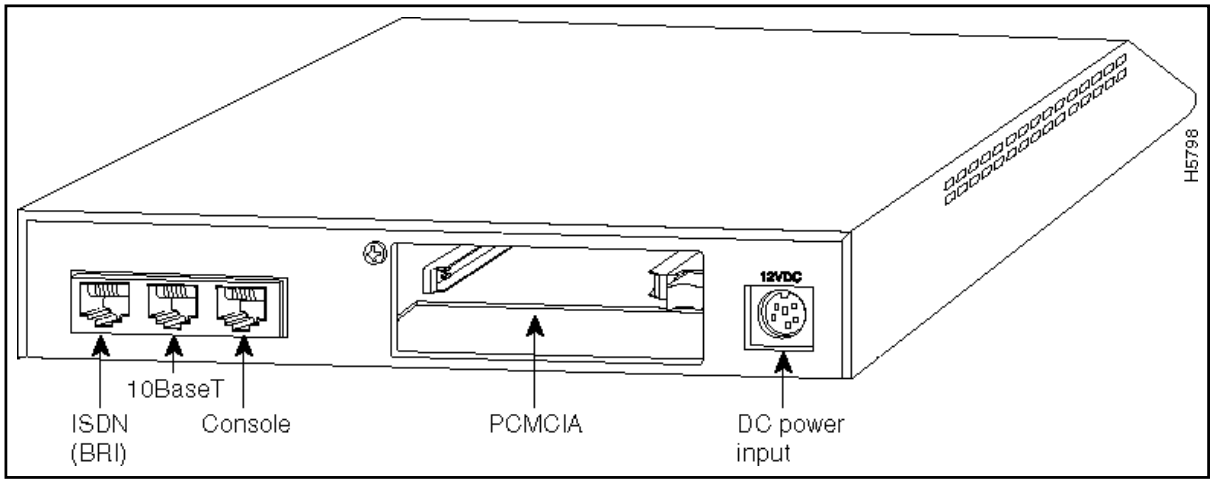

Figure 3.23 *CISCO 1003 - face arrière (publié avec l'aimable autorisation de CISCO)*

Type de réseau

Définissons tout d'abord le type de réseau auquel on se raccorde. En France, il peut s'agir de *vn2*, *vn3* ou *vn4*. On utilise pour cela la commande `isdn switch-type`:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#isdn switch-type vn3
Router(config)#^Z
Router#
```

Configuration de l'interface Ethernet

Utilisons les commandes `ip address` et `ip broadcast-address` pour configurer les paramètres liés à l'interface Ethernet :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Ethernet 0
Router(config-if)#description Interface sur le LAN
Router(config-if)#ip address 192.168.22.33 255.255.255.224
Router(config-if)#ip broadcast-address 192.168.22.63
Router(config-if)#^Z
Router#
```

Configuration de la route par défaut

Pour configurer la route par défaut, on indique un réseau par défaut par la commande `ip default-network`, et une route statique vers ce réseau par la commande `ip route`. Le réseau local du fournisseur est donc choisi comme réseau par défaut. La route statique pointe

vers l'adresse IP du routeur du fournisseur dans le réseau d'interconnexion :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip default-network 192.168.100.0
Router(config)#ip route 192.168.100.0 255.255.255.0 192.168.200.1
Router(config)#^Z
Router#
```

L'interface BRI du routeur appartient à un réseau d'interconnexion dont font partie les interfaces BRI des équipements des autres clients. Il faut donc, si on veut pouvoir les joindre, ajouter une route pour le réseau d'interconnexion, passant par le fournisseur.

On procède de la façon suivante :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 192.168.200.0 255.255.255.0 192.168.200.1
Router(config)#^Z
Router#
```

Configuration des adresses de l'interface BRI

Configurons les paramètres liés à l'interface BRI de la même façon qu'on a configuré les adresses liées à l'interface Ethernet :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface BRI 0
Router(config-if)#description Interface vers le fournisseur
Router(config-if)#ip address 192.168.200.2 255.255.255.0
Router(config-if)#ip broadcast-address 192.168.200.255
Router(config-if)#^Z
Router#
```

Configuration du protocole d'encapsulation

On définit le type d'encapsulation sur l'interface BRI à l'aide de la sous-commande d'interface `encapsulation`.

Pour demander au routeur d'utiliser plusieurs canaux B simultanément en suivant le standard PPP Multilink Protocol, on utilise la sous-commande d'interface `ppp multilink`, après avoir pris soin de vérifier que le routeur du fournisseur se conforme à cette norme. On doit alors définir la charge à partir de laquelle un nouveau canal est ouvert. La sous-commande d'interface `dialer load-threshold` permet de définir ce seuil. On lui fournit un numéro compris entre 1 et 256 pour définir la proportion de débit correspondant au seuil. Par exemple, la valeur 128 indique au routeur d'ouvrir un canal B supplémentaire quand le premier atteint 50 % de charge.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface BRI 0
Router(config-if)#encapsulation ppp
Router(config-if)#dialer load-threshold 128
Router(config-if)#ppp multilink
Router(config-if)#^Z
Router#
```

Configuration de l'authentification

Supposons que le fournisseur nous demande de configurer l'équipement avec une authentification de type CHAP. C'est le cas le plus répandu. Pour mettre en place ce type d'authentification, il nous faut un secret commun avec le serveur. Choisissons donc pour cela la chaîne de caractères `leSecret`. Le client et le fournisseur doivent de plus s'identifier par un nom avec CHAP. C'est le nom du routeur qui est utilisé pour cela. On va donc utiliser le nom `routeurClient` pour notre routeur et `routeurFournisseur` pour celui du fournisseur. On configure maintenant les noms et le secret à l'aide des commandes `hostname` et `username` :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname routeurClient
routeurClient(config)#username routeurFournisseur password leSecret
routeurClient(config)#^Z
routeurClient#
```

On applique alors le protocole d'authentification à l'interface BRI :

```
routeurClient#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
routeurClient(config)#interface BRI 0
routeurClient(config-if)#ppp authentication chap
routeurClient(config-if)#^Z
routeurClient#
```

Configuration du numéro d'appel

On définit une entrée de la *dialer map* de l'interface BRI pour indiquer l'association entre le numéro d'appel RNIS, l'adresse IP du routeur du fournisseur dans le réseau d'interconnexion et le nom de cet équipement distant pour la phase d'authentification CHAP :

```
RouterClient#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
routeurClient(config)#interface BRI 0
routeurClient(config-if)#dialer map ip 192.168.200.1 name routeurFournisseur 0123456789
routeurClient(config-if)#^Z
routeurClient#
```

Cela permet au routeur de savoir quel numéro composer quand un paquet à destination de l'Internet lui parvient.

Configuration des conditions d'appel

Souvent, seuls certains paquets sont autorisés à activer une connexion RNIS. Par exemple, on peut souhaiter que seuls les services TCP ou UDP aient la possibilité d'activer la liaison RNIS, mais pas les paquets ICMP, afin d'éviter par exemple que la connexion reste active plusieurs heures durant à cause d'une erreur de configuration ou de manipulation.

Pour cette raison, il faut systématiquement associer un numéro de *dialer-group* à l'interface BRI, avec la sous-commande d'interface `dialer-group`. On a alors le choix d'associer à ce *dialer-group* une *access-list* (commande `dialer-list`) décrivant les datagrammes autorisés à activer la connexion, ou d'indiquer directement les conditions d'activation. Le cas échéant, on saisit l'*access-list* avec la commande `access-list` qui est répétée pour chaque entrée de la liste.

Par exemple, pour autoriser tous les paquets IP à activer un canal, on va entrer la configuration suivante :

```
RouterClient#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
routeurClient(config)#interface BRI 0
routeurClient(config-if)#dialer-group 1
routeurClient(config-if)#exit
routeurClient(config)#dialer-list 1 protocol ip permit
routeurClient(config)#^Z
routeurClient#
```

Pour utiliser une *access-list*, on entre plutôt la configuration suivante :

```
RouterClient#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
routeurClient(config)#interface BRI 0
routeurClient(config-if)#dialer-group 1
routeurClient(config-if)#exit
routeurClient(config)#access-list 100 permit tcp any any
routeurClient(config)#access-list 100 permit udp any any
routeurClient(config)#dialer-list 1 list 100
routeurClient(config)#^Z
routeurClient#
```

On a donc défini une *access-list* au format étendu (les *access-list* étendues doivent avoir un numéro compris entre 100 et 199) qui n'accepte que les paquets TCP et UDP. On a appliqué avec `dialer-list` cette *access-list* à l'interface de *dialer-group* 1, c'est-à-dire à l'interface BRI. Nous verrons au chapitre 14 page 427 le format complet des *access-list* étendues.

Déconnexion automatique

On va maintenant définir le temps au bout duquel la liaison est coupée si rien n'a transité sur la ligne, à l'aide de la sous-commande d'interface `dialer idle-timeout`, qui prend un nombre de secondes en paramètre :

```
RouterClient#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
routeurClient(config)#interface BRI 0
routeurClient(config-if)#dialer idle-timeout 30
routeurClient(config-if)#^Z
routeurClient#
```

Activation du routage

Nous activons le routage RIP pour permettre au routeur de joindre les différents brins Ethernet internes.

Il ne faut évidemment pas faire d'annonce sur l'interface RNIS, sous peine de voir un canal B établi en permanence. On configure donc le routage ainsi :

```
RouterClient(config)#router rip
RouterClient(config-router)#network 192.168.22.0
RouterClient(config-router)#passive-interface BRI 0
RouterClient(config-router)#^Z
RouterClient#
```

Configuration complète

L'annexe A.1 page 458 présente le fichier de configuration d'un routeur CISCO 1003.

3.6 Liaison spécialisée numérique

3.6.1 Principe

Une liaison spécialisée numérique, aussi appelée ligne louée (*leased line*), est une liaison numérique permanente entre deux points, appelés extrémités. Ce service, fourni en France par France Télécom sous le nom de Transfix, est souvent désigné par l'expression « LS Transfix ». Il s'agit de la réservation d'un circuit de données numérique permanent entre deux centraux téléphoniques, ainsi que de la mise en place de deux boucles locales entre ces centraux, les raccordant aux extrémités de la liaison.

Le débit d'une liaison Transfix peut aller de quelques dizaines de Kbits/s à plusieurs Mbits/s, mais les fournisseurs Internet se limitent pour la plupart aux débits de 64, 128 et 512 Kbits/s. Quelques fournisseurs proposent des raccordements à 2 ou 34 Mbits/s. À la différence des accès à la demande tels que le RTC ou RNIS, la liaison spécialisée est disponible à tout moment, et ne nécessite donc pas d'étape d'établissement de la connexion. Ainsi, l'opérateur télécom ne facture pas cette liaison suivant la durée d'utilisation, notion qui n'a ici plus de sens, mais en fonction de la distance entre les deux extrémités. Il faut bien comprendre qu'un raccordement à plusieurs Mbits/s ne permet pas d'obtenir de tels débits entre le réseau du client d'un fournisseur européen et par exemple celui d'un serveur situé aux États-Unis.

Les fournisseurs français de qualité offrent aujourd'hui entre 6 et 10 Ko/s de débit utile au niveau TCP, à travers leurs lignes transatlantiques. En fin de semaine ou en pleine nuit, les débits offerts peuvent être beaucoup plus importants mais il ne faut pas en attendre plus en pleine journée, hors périodes estivales.

Néanmoins, plusieurs raisons peuvent amener à choisir un débit important avec le fournisseur, par exemple 2 Mbits/s :

- Le réseau du fournisseur, qui relie l'ensemble de ses clients entre eux, est dans la grande majorité des cas un réseau très rapide et peu engorgé. Si on choisit un fournisseur qui possède beaucoup de clients, on peut alors profiter d'excellents débits avec eux. C'est notamment le cas du réseau de la recherche français, Renater, qui interconnecte des plaques régionales avec un débit de 34 Mbits/s ; les universités raccordées à ces plaques régionales, souvent à 2 Mbits/s, voire même 34 Mbits/s, vont obtenir des débits impressionnants entre elles.
- Le fournisseur possède parfois, sur des serveurs installés chez lui, des archives des serveurs de fichiers majeurs remises à jour régulièrement. C'est par exemple le cas d'OLÉANE qui propose, sur `ftp.oleane.net`, différentes archives de logiciels clients et serveurs fournis en domaine public sur l'Internet. Il est alors très utile de posséder une liaison spécialisée haut débit avec le fournisseur, pour profiter pleinement de ces archives locales.
- Le transfert des articles émis sur les forums depuis le fournisseur vers un serveur NNTP sur le site du client peut nécessiter énormément de débit si ce dernier a demandé à recevoir de nombreux groupes. Notons que 100 et 150 Kbits/s sont nécessaires en permanence pour transférer le flux complet d'articles si on s'abonne à l'ensemble des forums.

Le protocole utilisé sur la ligne afin d'encapsuler les datagrammes IP peut être PPP ou HDLC, au choix du fournisseur.

3.6.2 Équipements

La figure 3.24 page 125 présente les différents équipements qui entrent en jeu lors d'un raccordement par LS Transfix :

- Chez le client, on trouve un routeur de proximité connecté par une interface Ethernet au réseau local, et muni d'une interface série synchrone raccordée à un modem bande de base posé par l'opérateur télécom ; cette interface synchrone est souvent de type X24/V11 ou V24/V35. Les caractéristiques mécaniques de cette dernière sont de deux types : interface V35 français et interface V35 US.

  La jonction série synchrone qui relie les deux équipements est ainsi composée de deux câbles, comme le montre la figure 3.25 page 125 :

  - un câble que le client doit se procurer auprès du distributeur du routeur spécialisé ; il est connecté d'un côté au boîtier du routeur, et propose par exemple de l'autre

côté une interface de type V35 français mâle ;
- un câble fourni par l'opérateur télécom, connecté d'un côté au boîtier du modem bande de base et proposant par exemple de l'autre côté une interface de type V35 français femelle.

Il faut donc impérativement que le client contacte l'opérateur avant de se procurer le premier de ces deux câbles, pour s'assurer que les interfaces correspondent. En effet, le coût de cet accessoire constitue souvent une fraction non négligeable du coût global du routeur de proximité.

- Le fournisseur possède le plus souvent un ou plusieurs routeurs disposant chacun de plusieurs dizaines d'interfaces série synchrones ; il y relie alors les modems bande de base apportés par l'opérateur télécom pour chacun de ses clients.

  D'autres configurations existent. Par exemple, lorsque le fournisseur possède un très grand nombre de LS Transfix, l'opérateur télécom met en place un point d'accès transfix (PAT Transfix), qui possède en entrée une ou plusieurs interfaces de type E1, chacune multiplexant temporellement jusqu'à 30 canaux à 64 Kbits/s. Le PAT se charge ainsi d'extraire les différents signaux correspondant aux différentes LS et les propose à travers autant d'interfaces synchrones. Le fournisseur se contente de raccorder ces interfaces à ses routeurs.

3.6.3 Plan d'adressage

Dans le cadre d'un raccordement par LS Transfix, le fournisseur doit informer son client des adresses IP utilisées pour les interfaces du côté ligne de transmission des deux routeurs en jeu. Il réserve pour cela un réseau d'interconnexion. Mais à la différence du réseau d'interconnexion que nous avions étudié lors du raccordement par RNIS avec routeur dédié en section 3.5.5 page 111, ce réseau est propre à chaque client. Remarquons en effet que le routeur du fournisseur possède ici une interface par client, alors que dans la section 3.5.5, le routeur disposait d'une interface pour tous ses clients, ce qui justifie ce nouveau type de réseau d'interconnexion.

Pour attribuer des réseaux d'interconnexion avec ses clients, le fournisseur choisit donc un réseau de classe C, qu'il découpe en sous-réseaux par la méthode exposée section 2.2.2 page 37. Chacun de ces sous-réseaux fait office de sous-réseau d'interconnexion, et comporte donc :

- une adresse IP pour le fournisseur,
- une adresse IP pour le client,
- une adresse de réseau,
- une adresse de diffusion.

Il faut donc quatre adresses IP au moins dans chaque sous-réseau. On réserve donc 2 bits pour la partie nœuds, et il reste 6 bits pour la partie sous-réseaux dans le dernier octet du masque de sous-réseaux. Le masque se représente donc ainsi en binaire :

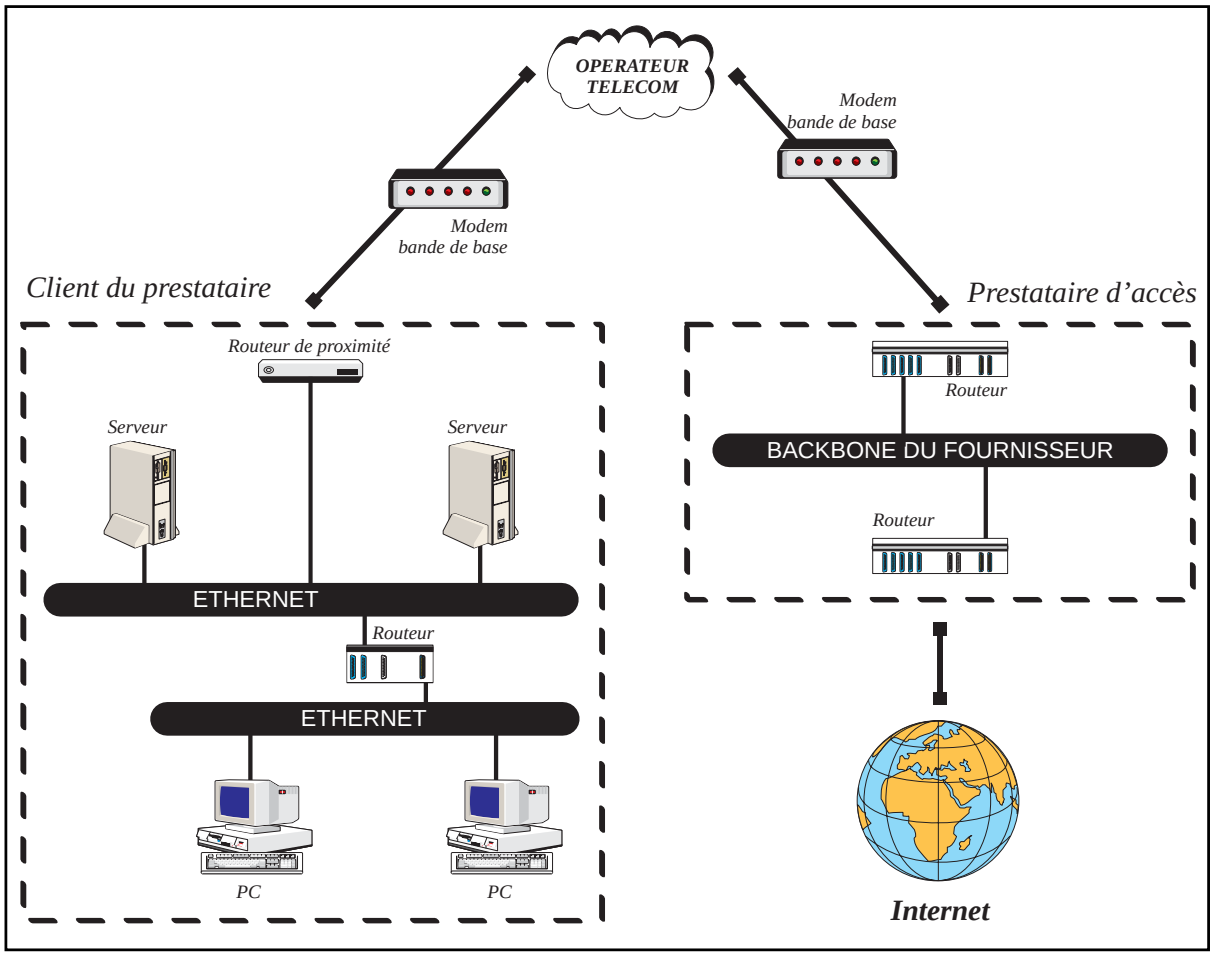

Figure 3.24 *Ligne spécialisée avec routeur de proximité*

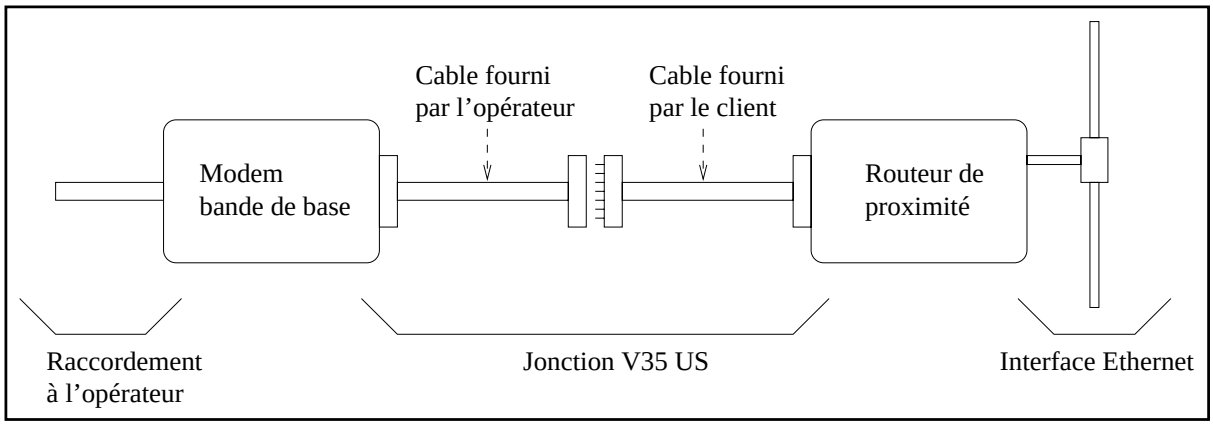

Figure 3.25 *Le raccordement au modem bande de base*

`11111111.11111111.11111111.11111100`.
Exprimé en décimal, il s'écrit `255.255.255.252`.

Supposons par exemple qu'un fournisseur ait choisi le réseau `192.168.205.0`. Le tableau 3.2 indique quelques réseaux d'interconnexion avec les clients de ce fournisseur.

Client	**Adresse IP**	**type**
	`192.168.205.0/30`	réseau réservé
client A	`192.168.205.4/30`	réseau d'interconnexion pour A
	`192.168.205.5`	interface synchrone du routeur du client
	`192.168.205.6`	interface synchrone du routeur du fournisseur
	`192.168.205.7`	adresse de diffusion
client B	`192.168.205.8/30`	réseau d'interconnexion pour B
	`192.168.205.9`	interface synchrone du routeur du client
	`192.168.205.10`	interface synchrone du routeur du fournisseur
	`192.168.205.11`	adresse de diffusion
client C	`192.168.205.12/30`	réseau d'interconnexion pour C
	`192.168.205.13`	interface synchrone du routeur du client
	`192.168.205.14`	interface synchrone du routeur du fournisseur
	`192.168.205.15`	adresse de diffusion
client D	`192.168.205.16/30`	réseau d'interconnexion pour D
	`192.168.205.17`	interface synchrone du routeur du client
	`192.168.205.18`	interface synchrone du routeur du fournisseur
	`192.168.205.19`	adresse de diffusion
client E	`192.168.205.20/30`	réseau d'interconnexion pour E
	`192.168.205.21`	interface synchrone du routeur du client
	`192.168.205.22`	interface synchrone du routeur du fournisseur
	`192.168.205.23`	adresse de diffusion

Tableau 3.2 *Réseaux d'interconnexion*

La figure 3.26 page suivante présente le plan d'adressage du raccordement du client E.

3.6.4 Mise en place du routage

De même qu'avec les autres types de raccordement, le fournisseur se charge d'annoncer le réseau de classe B ou C du client sur l'Internet par le protocole BGP-4.

Mais à la différence des autres types de raccordement, il arrive que le fournisseur demande à son client d'annoncer lui-même, par l'intermédiaire de son routeur de proximité, l'accessibilité de ses réseaux par un protocole de routage interne ; il s'agit de la deuxième politique de routage évoquée section 2.6.3 page 52. Dans ce dernier cas, le fournisseur annonce aussi à chacun de ses clients l'ensemble des routes de ses autres clients.

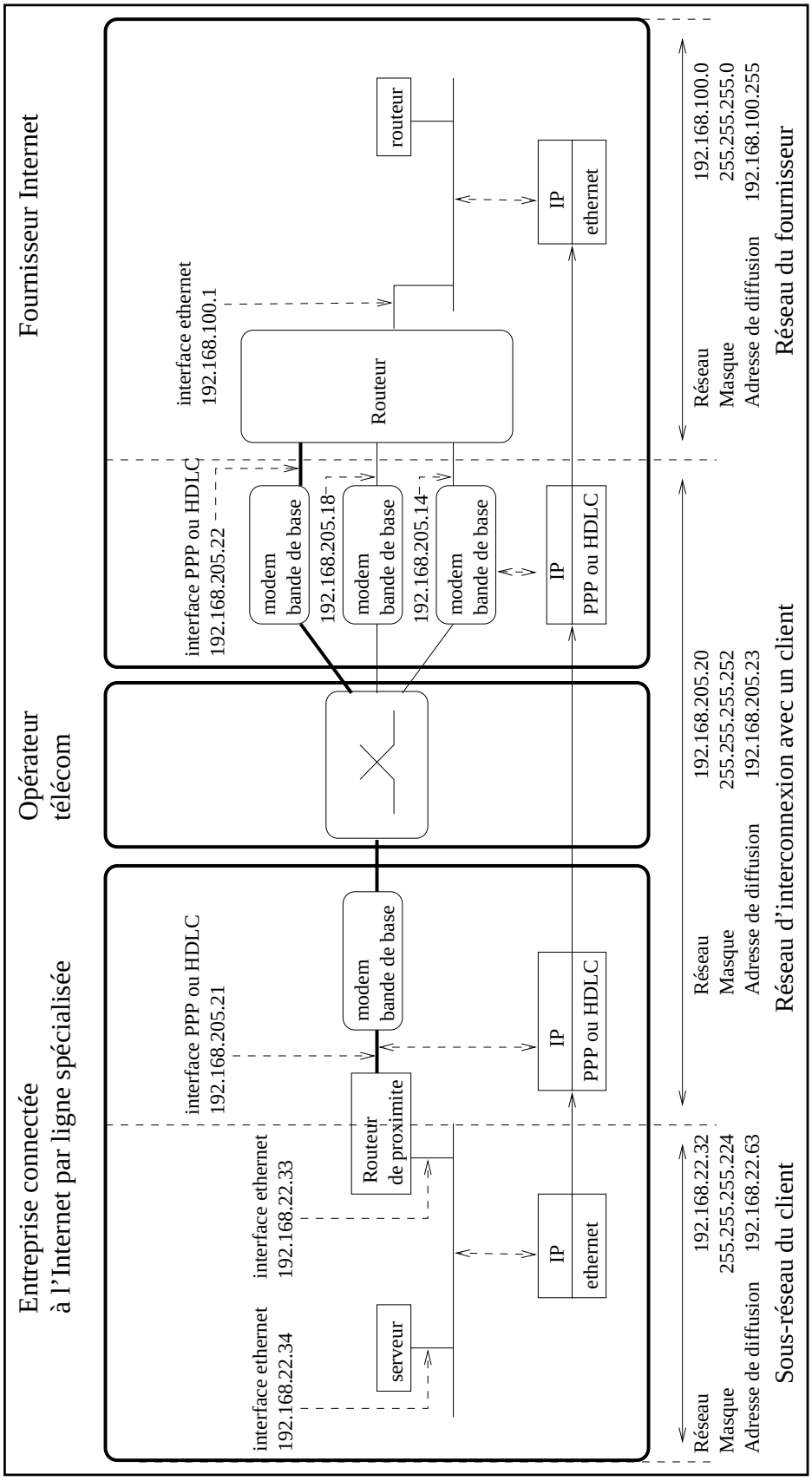

Figure 3.26 *Ligne spécialisée avec routeur de proximité*

3.6.5 Particularités du raccordement par ligne spécialisée

La liaison spécialisée constitue le seul type de raccordement qui puisse permettre de mettre en place efficacement sur un site des services Internet tels qu'un serveur World Wide Web ou un serveur FTP de transfert de fichiers. Certes, il est techniquement possible de mettre en place un serveur Web sur un site raccordé par Transpac car le fournisseur est à même d'établir la connexion au moment où un client Web sur l'Internet désire y accéder, mais le temps de connexion important, dû à l'établissement du circuit virtuel X25 qui vient augmenter le temps de traversée du réseau Internet, est incompatible avec une utilisation agréable dans le cadre de services destinés à une large diffusion.

D'autre part, l'opérateur télécom qui pose la ligne spécialisée entre la salle machine du fournisseur et celle de son client a, pour ces deux extrémités, un seul interlocuteur principal, en l'occurrence le client de l'opérateur télécom. Ainsi, deux cas se présentent :

1. Le fournisseur est le client de l'opérateur. Ce cas est souvent imposé par le fournisseur Internet. Il en tire différents avantages :
   - cela lui permet d'être le contact privilégié de l'opérateur. Ils maîtrisent tous les deux la technique, cela est très utile lors de la phase de mise en place et de tests de la liaison ;
   - en cas de problème sur la ligne, l'opérateur prévient directement le fournisseur ;
   - lorsqu'un client résilie son abonnement, la ligne n'est pas fermée, le fournisseur demande auprès de l'opérateur la migration de la boucle locale côté client vers un nouveau client, ce qui coûte moins cher que de clore une liaison pour en ouvrir une autre ;
   - le fournisseur est maître des tarifs : il peut décider d'un tarif forfaitaire pour les clients situés jusqu'à une certaine distance d'un de ses points de présence. Ainsi, il peut à tout moment déplacer ce point de présence au barycentre de ses clients pour minimiser le coût global, sans changer les tarifs d'aucun client.
2. Le client du fournisseur est le client de l'opérateur. Le fournisseur évite ce type de situation sauf quand la distance entre son point de présence et les locaux de son client est très importante, car alors, la ligne spécialisée coûte beaucoup plus cher que d'ordinaire.

   L'avantage principal pour le client est de diminuer le coût du changement de fournisseur Internet : en étant client de l'opérateur, il lui suffit de demander la migration de la boucle locale chez le fournisseur qu'il veut quitter vers le point de présence du fournisseur qu'il rejoint, l'opération lui coûte ainsi moins cher que la création d'une nouvelle ligne spécialisée.

3.6.6 Quelques routeurs de proximité

De nombreux constructeurs tels que 3COM, CISCO ou Welfleet, proposent des routeurs capables de raccorder une ligne spécialisée à un réseau local.

Routeurs CISCO 1005 et 2503

CISCO propose des modèles d'entrée de gamme économiques offrant une ou deux interfaces série synchrones et une interface de réseau local de type Token Ring, Ethernet ou FDDI. Par exemple, le CISCO 1005 est un modèle d'entrée de gamme disposant d'une interface série synchrone capable de fonctionner jusqu'à 2 Mbits/s. Le modèle CISCO 2503 présenté sur la figure 3.27 est un routeur évolutif qui dispose d'une interface BRI pour un raccordement par RNIS, ainsi que de deux interfaces série synchrones pour un raccordement par ligne spécialisée. Ainsi, avec un routeur CISCO 2503, on peut commencer par s'abonner à un service d'accès Internet par RNIS, puis, avec le même matériel, faire évoluer son abonnement vers une ligne spécialisée.

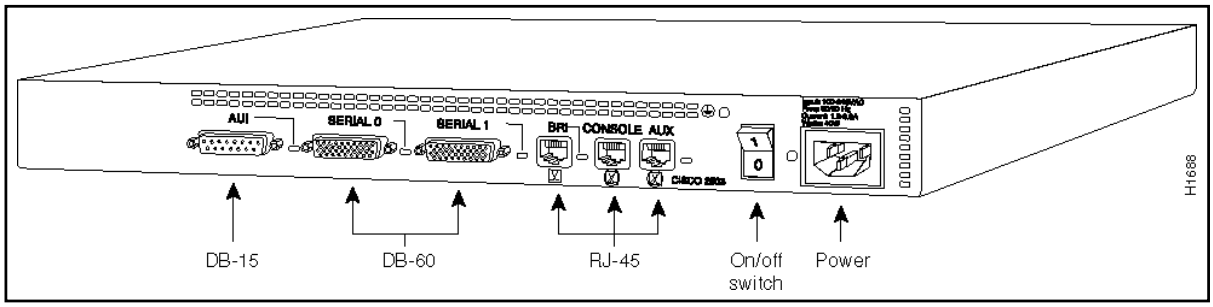

Figure 3.27 *CISCO 2503 - face arrière (publié avec l'aimable autorisation de CISCO)*

Examinons les particularités de la configuration d'un routeur CISCO disposant d'une interface série synchrone. Le lecteur est prié de se reporter section 3.3 page 73 pour une introduction à la configuration d'un routeur CISCO.

Configuration de l'interface Ethernet

Utilisons les commandes `ip address` et `ip broadcast-address` afin de configurer les paramètres liés à l'interface Ethernet :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Ethernet 0
Router(config-if)#description Interface sur le LAN
Router(config-if)#ip address 192.168.22.33 255.255.255.224
Router(config-if)#ip broadcast-address 192.168.22.63
Router(config-if)#^Z
Router#
```

Configuration de la route par défaut

Pour configurer la route par défaut, on indique un réseau par défaut par la commande `ip default-network` et une route statique vers ce réseau par la commande `ip route`. Le

réseau du fournisseur est donc choisi comme réseau par défaut. La route statique pointe vers l'adresse IP du routeur du fournisseur dans le réseau d'interconnexion :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip default-network 192.168.100.0
Router(config)#ip route 192.168.100.0 255.255.255.0 192.168.205.22
Router(config)#^Z
Router#
```

Configuration des adresses de l'interface série

Configurons les paramètres liés à l'interface série de la même façon qu'on a configuré les adresses liées à l'interface Ethernet :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0
Router(config-if)#description Interface sur ligne specialisee
Router(config-if)#ip address 192.168.205.21 255.255.255.252
Router(config-if)#ip broadcast-address 192.168.205.23
Router(config-if)#^Z
Router#
```

Configuration du protocole d'encapsulation et du débit

L'encapsulation par défaut sur la ligne spécialisée est de type IP sur HDLC. Pour une encapsulation de type PPP, il faut utiliser la sous-commande d'interface `encapsulation ppp`. Le débit se définit avec la sous-commande d'interface `bandwidth`. Par exemple, pour une ligne spécialisée à 64 Kbits/s, il faut configurer le routeur comme suit :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0
Router(config-if)#bandwidth 64
Router(config-if)#^Z
Router#
```

Activation du routage

Nous activons le routage RIP pour permettre au routeur de joindre les différents brins Ethernet internes :

```
Router(config)#router rip
Router(config-router)#network 192.168.22.0
Router(config-router)#^Z
Router#
```

Configuration complète

L'annexe A.2 page 459 présente le fichier de configuration d'un routeur CISCO 1005.

3.7 X25/Transpac

3.7.1 Principe

Le réseau Transpac est un réseau à commutation de paquets. L'interface avec le réseau Transpac se conforme au protocole X25, et permet ainsi l'établissement de circuits virtuels de données, souvent désignés par le terme CV, entre les différents sites abonnés.

Ces derniers sont désignés par des adresses de type X.121 qui s'écrivent sous forme de suites de chiffres décimaux. Le plan d'adressage X.121 défini par l'ITU-T découpe les adresses X.121 en deux champs :

- le DNIC (Data Network Identification Code) identifie l'opérateur de manière unique. Il est formé de deux parties : le code du pays, sur trois chiffres, et le numéro d'opérateur X25 dans le pays, sur un chiffre ;
- le NTN (National Terminal Number) identifie le client de manière unique pour un opérateur national donné. Ce champ peut contenir jusqu'à dix chiffres.

Le plan d'adressage X.121 est représenté sur la figure 3.28.

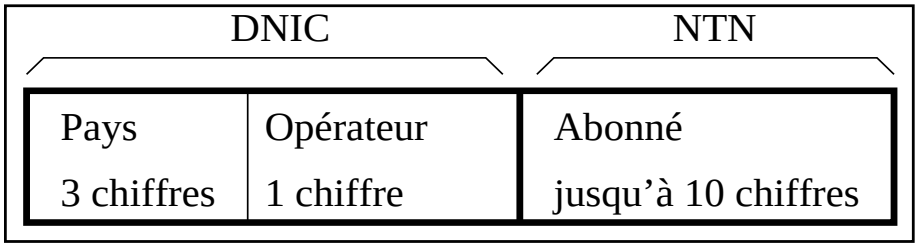

Figure 3.28 *Plan d'adressage X.121*

Sur le circuit virtuel, le fournisseur Internet et ses clients utilisent une encapsulation de type IP sur X25.

Le coût d'utilisation de ce réseau dépend notamment de la quantité d'informations transférées, et non pas de la distance entre les deux sites communiquant : c'est le site qui a demandé l'ouverture d'un circuit qui prend en charge le coût total de la transmission, quel qu'en soit le sens. Mais il est possible d'émettre une demande d'établissement d'un circuit virtuel avec taxation au demandé, c'est-à-dire d'imputer le coût de transmission à l'appelé, et non à l'appelant. Cette fonctionnalité permet à un fournisseur de services d'ouvrir un circuit à destination d'un de ses clients quand un datagramme IP provenant de l'Internet transite par son réseau local avec une adresse IP destination chez le client en question. L'ouverture du circuit

n'étant pas à la charge unique du client, ce dernier peut ainsi mettre en place des serveurs sur son site : la connexion Transpac qui est intermittente apparaît donc ici comme un raccordement permanent à l'Internet.

3.7.2 Accès au réseau

Plusieurs moyens d'accès au réseau Transpac existent :

- Les accès synchrones permettent de se connecter à un fournisseur avec le protocole IP sur X25 pour raccorder un réseau local comme on le ferait avec PPP à travers un modem ou avec HDLC sur une ligne spécialisée ; il existe plusieurs types d'accès synchrones au réseau Transpac :
  - *accès direct* - c'est l'accès privilégié. Il consiste à mettre en place une ligne spécialisée entre le site et le plus proche centre technique Transpac. Cette ligne peut avoir un débit allant jusqu'à 2 Mbits/s. La jonction est alors de type X21bis/V28 au niveau physique, HDLC-LAPB au niveau liaison de données, et X25 au niveau réseau ;
  - *accès à travers le RTC et RNIS sur canal D jusqu'à 9600 bits/s ou RNIS sur canal B à 64 Kbits/s, à travers des EBS (Entrée banalisée synchrone) ou ERS (Entrée réservée synchrone).* Notamment, en composant le 3603, avec un modem V32, on se connecte par le RTC à 9600 bits/s à une EBS, et on atteint par le 0836063232 une EBS multi vitesse. Par RNIS, il faut composer le 0836086464 pour un accès EBS-64, c'est-à-dire sur canal B.
- Les accès asynchrones permettent d'entrer sur le réseau Transpac à partir de terminaux bas de gamme qui ne disposent pas du protocole X25. Cela consiste à utiliser un modem pour accéder par le RTC à un PAD (Packet Assembler Disassembler) X3 qui assure la traduction entre le protocole asynchrone au niveau caractère X28 et le protocole X29 permettant à un PAD de dialoguer sur le réseau X25 avec le site du fournisseur. Ce procédé permet une connexion en mode caractères, par exemple pour tranférer du courrier électronique par UUCP ou pour permettre un accès distant de type émulateur de terminal VT100. La figure 3.29 indique les différents protocoles en jeu dans ce type de connexion.

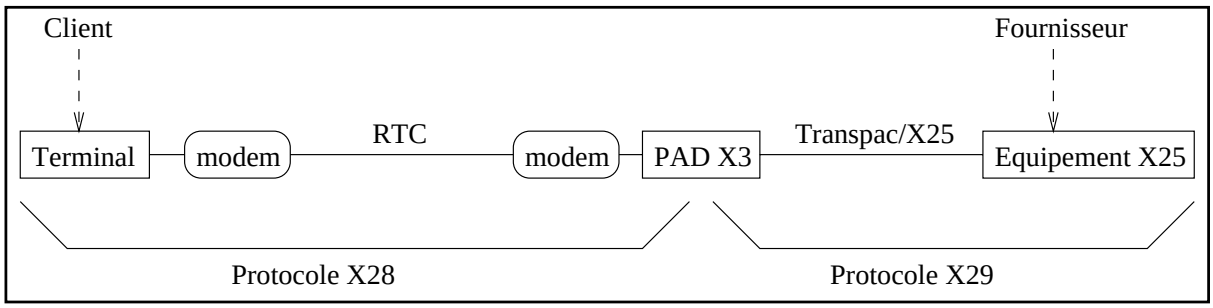

Figure 3.29 *Accès au fournisseur Internet à travers un PAD*

Par la suite, nous nous contenterons de traiter l'accès direct synchrone qui est le plus répandu dans le cadre de l'accès des professionnels à l'Internet via le réseau Transpac.

3.7.3 Équipements

La figure 3.30 présente les différents équipements qui entrent en jeu lors d'une connexion par X25 sur Transpac : le client et le fournisseur relient un routeur connecté à leur réseau local par une jonction synchrone de type X21bis/V28 à l'extrémité de la ligne spécialisée qui les raccorde à un centre d'opérations de Transpac.

Évidemment, le fournisseur ne possède qu'un équipement connecté sur la ligne spécialisée avec Transpac, pour l'ensemble de ses clients. En effet, sur un seul accès, jusqu'à 4095 voies logiques peuvent être actives simultanément.

Le client se voit poser une ligne spécialisée, le lecteur est donc prié de se reporter à la section 3.6.2 page 123 pour plus de renseignements sur la jonction et les câbles nécessaires pour ce type de raccordement.

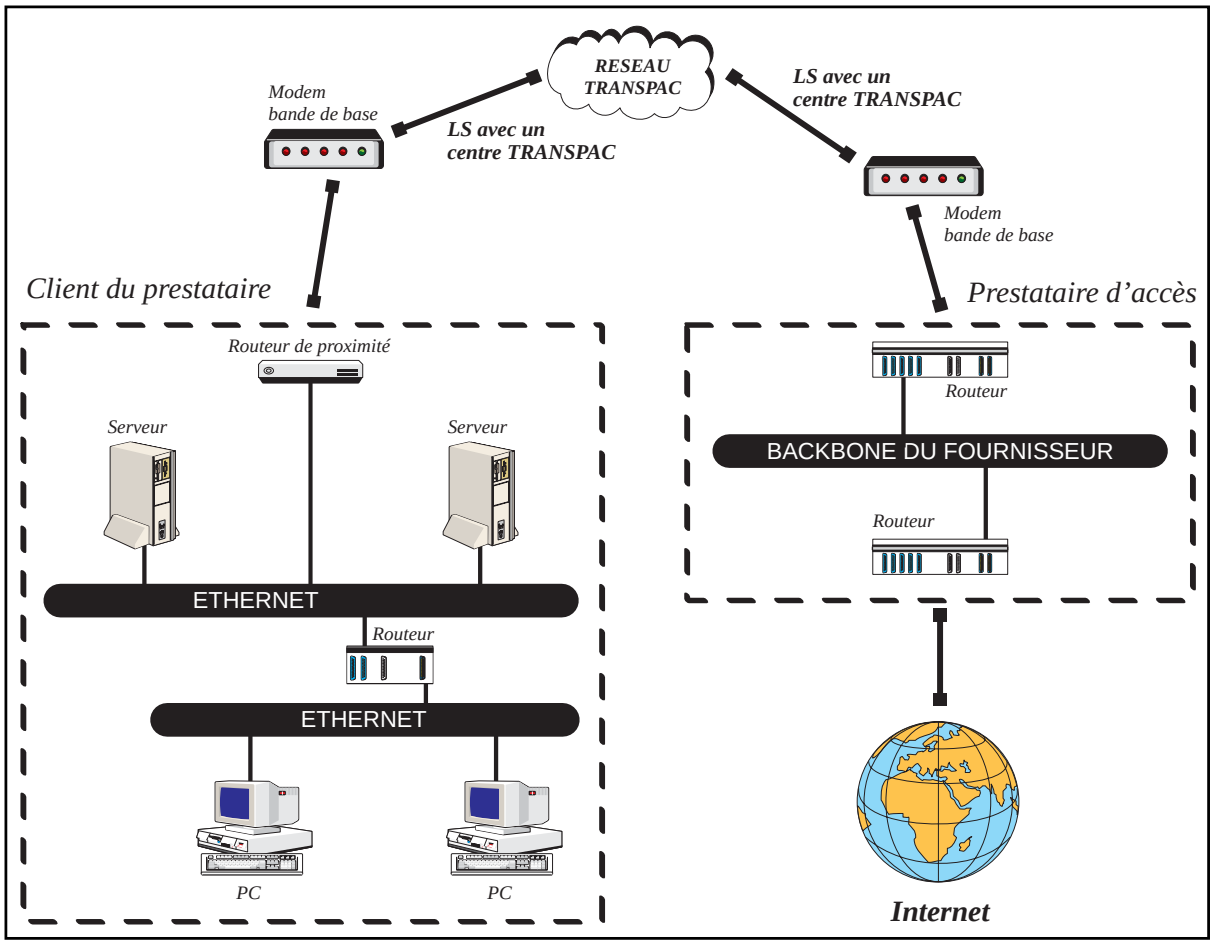

Figure 3.30 *Accès au fournisseur Internet par Transpac : topologie*

3.7.4 Plan d'adressage

De même que lors de l'analyse du plan d'adressage d'un raccordement par ligne spécialisée étudié section 3.5.5 page 111, le fournisseur dispose ici d'un routeur muni d'une unique interface physique avec le réseau Transpac pour les connexions de l'ensemble de ses clients, il doit donc définir, de même qu'en section 3.5.5, un réseau d'interconnexion partagé par l'ensemble de ses clients.

Le fournisseur attribue donc un unique réseau de classe C accueillant les adresses IP des interfaces IP/X25 des clients et celle de son routeur.

Prenons pour exemple le réseau d'interconnexion `192.168.210.0`. L'adresse réseau du fournisseur est `192.168.100.0` et son adresse X121 est 17511110.

Reprenons ainsi l'exemple de réseau client utilisé au chapitre 2 pour présenter les masques de sous-réseaux. Ce réseau porte le numéro `192.168.22.0` et le routeur connecté à l'Internet possède l'adresse IP `192.168.22.33`. Nous l'appellerons client A et l'adresse X121 attribuée par Transpac pour ce site est 17511111. Considérons aussi que le fournisseur possède trois autres clients, B, C et D, ayant souscrit au service d'accès Internet par Transpac. Les couples réseau de classe C et adresse X121 qui leur ont été attribués sont les suivants :

- `192.168.23.0` et 17511112,
- `192.168.24.0` et 17511113,
- `192.168.25.0` et 17511114.

Le tableau 3.3 récapitule les réseaux de classe C en jeu.

Entité	**Adresse réseau**	**masque de sous-réseaux**
Client A	`192.168.22.0`	`255.255.255.224`
Client B	`192.168.23.0`	N.C.
Client C	`192.168.24.0`	N.C.
Client D	`192.168.25.0`	N.C.
Fournisseur	`192.168.100.0`	`255.255.255.0` (sans découpage en sous-réseaux)
Réseau d'interconnexion	`192.168.210.0`	`255.255.255.0` (sans découpage en sous-réseaux)

Tableau 3.3 *Réseaux en jeu lors du raccordement par X25*

Les adresses IP des différentes interfaces des équipements en jeu lors des connexions des clients sont décrites par le tableau 3.4 page 136.

La figure 3.31 page suivante indique les différentes interfaces en jeu dans le cadre du raccordement du client A, ainsi que leurs adresses IP.

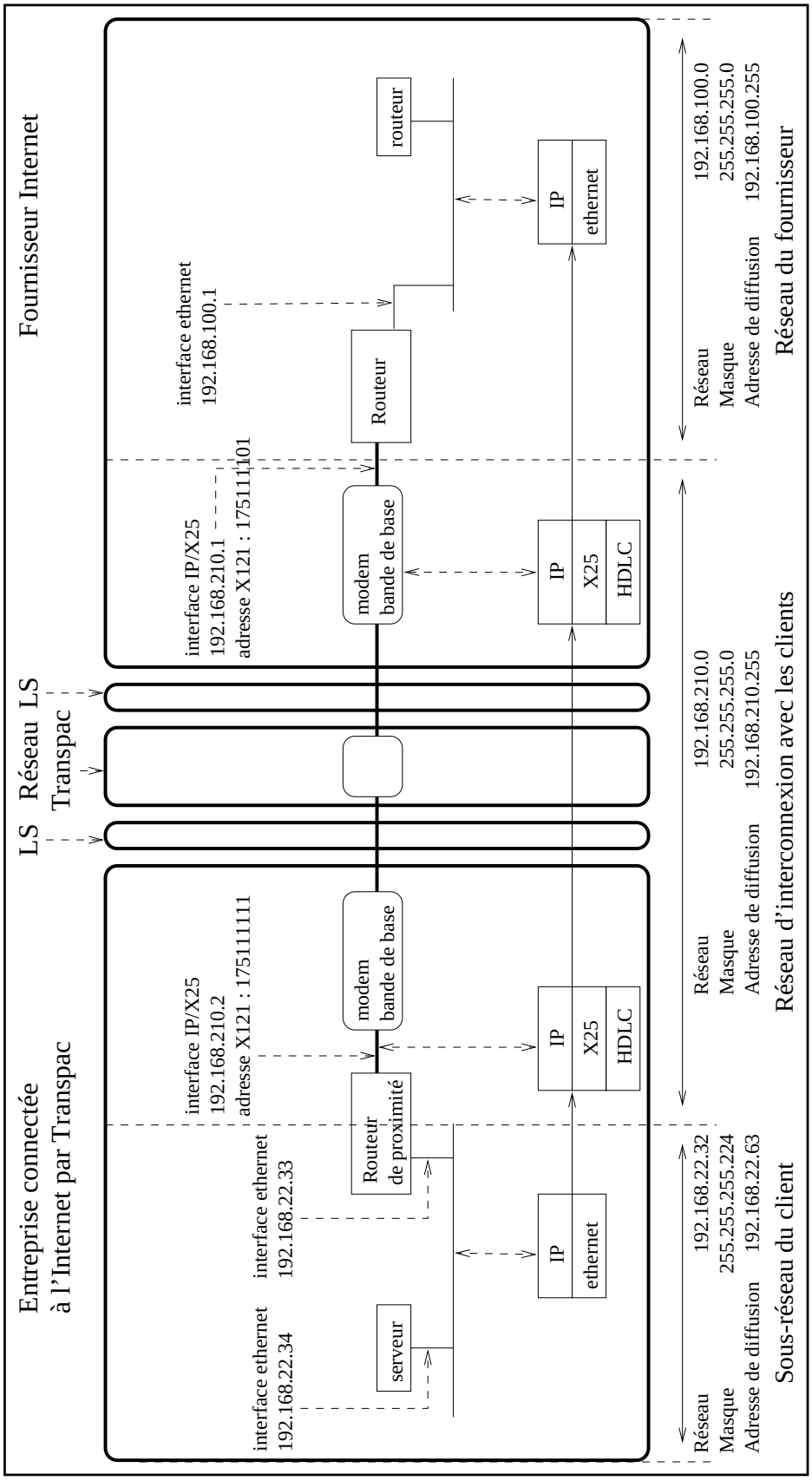

Figure 3.31 *Accès au fournisseur Internet par Transpac : plan d'adressage*

Entité	**Interface**	**Adresse IP** de sous-réseaux	**Adresse X121**
Client A	Ethernet	`192.168.22.33`	
	IP/X25	`192.168.200.2`	175111111
Client B	Ethernet	`192.168.23.50`	
	IP/X25	`192.168.200.3`	175111121
Client C	Ethernet	`192.168.24.91`	
	IP/X25	`192.168.200.4`	175111131
Client D	Ethernet	`192.168.25.72`	
	IP/X25	`192.168.200.5`	175111141
Fournisseur	Ethernet	`192.168.100.1`	
	IP/X25	`192.168.200.1`	175111101

Tableau 3.4 *Adresses IP des équipements*

3.7.5 Mise en place du routage

De même qu'avec les autres types de raccordements, le fournisseur se charge d'annoncer, sur l'Internet, le réseau de classe B ou C du client par le protocole BGP-4.

À la différence du raccordement par ligne spécialisée, il n'y a pas d'annonce de routage entre le client et le fournisseur car cela nécessiterait d'établir un circuit virtuel même si aucun trafic n'était demandé, et un coût supplémentaire et inutile serait ainsi imputé au client.

Un réseau d'interconnexion unique pour tous les clients est utilisé dans le cadre du raccordement par Transpac. On retrouve ainsi le problème décrit section 3.5.5 page 114 lors de l'étude de la mise en œuvre du routage pour une connexion RNIS avec routeur spécialisé, le lecteur est donc prié de s'y référer.

3.7.6 Commutateurs X25 et routeurs

Le raccordement X25 peut sembler plus complexe à mettre en place que les autres types de raccordements à l'Internet. Ce sont en fait, dans la grande majorité des cas, des sociétés déjà connectées à Transpac pour d'autres raisons et pas nécessairement pour le transport de datagrammes IP, qui sont amenées à se raccorder à l'Internet par ce biais. Cela leur permet, à moindre coût, d'établir une liaison avec un fournisseur : il n'y a pas à souscrire de nouvel abonnement auprès de Transpac. Par contre, les équipements dont elles disposent, notamment les commutateurs X25, ne sont pas systématiquement dotés de la possibilité d'encapsuler des paquets IP sur les liaisons de ce type.

Motorola, SAT, CISCO et bien d'autres constructeurs proposent des équipements qui incluent la gestion des protocoles nécessaires à l'encapsulation des datagrammes IP sur le réseau Transpac afin de se connecter à un fournisseur Internet. Ces équipements disposent donc, en plus de leur interface X21bis/V28, d'une interface de réseau local de type Ethernet ou autre.

Routeur CISCO 1003

Le routeur CISCO 1003 est un routeur d'entrée de gamme qui dispose d'une interface série synchrone permettant de connecter la ligne spécialisée raccordant le site au centre Transpac le plus proche.

Il sait encapsuler des datagrammes IP sur la couche X25 niveau 3. Nous avons déjà présenté ses caractéristiques physiques sur les figures 3.22 page 117 et 3.23 page 118.

Examinons les particularités de la configuration IP sur X25. Le lecteur est prié de se reporter section 3.3 page 73 pour une introduction à la configuration d'un routeur CISCO.

Configuration de l'interface Ethernet

À l'aide des commandes `ip address` et `ip broadcast-address`, configurons les paramètres liés à l'interface Ethernet :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Ethernet 0
Router(config-if)#description Interface sur le LAN
Router(config-if)#ip address 192.168.22.33 255.255.255.224
Router(config-if)#ip broadcast-address 192.168.22.63
Router(config-if)#^Z
Router#
```

Configuration de la route par défaut

Pour configurer la route par défaut, on indique un réseau par défaut par la commande `ip default-network` et une route statique vers ce réseau par la commande `ip route`. Le réseau du fournisseur est donc choisi comme réseau par défaut. La route statique pointe vers l'adresse IP du routeur du fournisseur dans le réseau d'interconnexion :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip default-network 192.168.100.0
Router(config)#ip route 192.168.100.0 255.255.255.0 192.168.210.1
Router(config)#^Z
Router#
```

L'interface X25 du routeur appartient à un réseau d'interconnexion dont font partie les interfaces X25 des équipements des autres clients. Il faut donc, si on veut pouvoir les joindre, ajouter une route pour le réseau d'interconnexion, passant par le fournisseur :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 192.168.210.0 255.255.255.0 192.168.210.1
Router(config)#^Z
Router#
```

Configuration des adresses de l'interface série

Configurons les paramètres liés à l'interface série de la même façon qu'on a configuré les adresses liées à l'interface Ethernet.

Notamment, on définit le débit de l'interface, 64 Kbits/s dans notre exemple :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0
Router(config-if)#description Interface vers Transpac
Router(config-if)#ip address 192.168.210.2 255.255.255.0
Router(config-if)#ip broadcast-address 192.168.210.255
Router(config-if)#bandwidth 64
Router(config-if)#^Z
Router#
```

Configuration du protocole d'encapsulation

Le protocole X25 est utilisé sur la ligne série. Il va donc falloir l'indiquer au routeur, ainsi que différents paramètres liés aux niveaux 2 (LAP-B) et 3 (X25 niveau réseau) du protocole.

Il est de règle de modifier le moins possible les paramètres niveau 2 pour en garder les valeurs par défaut, mais une modification des paramètres niveau 3 est souvent nécessaire.

Le tableau 3.5 regroupe les principaux paramètres X25 niveau 2 et le tableau 3.6 regroupe les principaux paramètres X25 niveau 3.

fonction	commande	exemple d'utilisation
adresse X121	x25 address	x25 address 175111111
délai de retransmission niveau 2 (en millisecondes)	lapb T1	lapb T1 150

Tableau 3.5 *Paramètres X25 niveau 2*

fonction	commande	exemple d'utilisation
taille de fenêtre niveau 3 en entrée	x25 win	x25 win 7
taille de fenêtre niveau 3 en sortie	x25 wout	x25 wout 7
taille maximum d'un paquet en entrée	x25 ips	x25 ips 512
taille maximum d'un paquet en sortie	x25 ops	x25 ops 512

Tableau 3.6 *Paramètres X25 niveau 3*

Les paramètres niveau 3, en l'occurrence `win`, `wout`, `ips` et `ops`, doivent être définis en concordance avec la configuration du commutateur X25 du centre Transpac.

De plus, on sélectionne le type d'encapsulation sur cette interface avec la sous-commande d'interface `encapsulation` :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0
Router(config-if)#encapsulation x25
Router(config-if)#x25 address 175111111
Router(config-if)#lapb T1 150
Router(config-if)#x25 win 7
Router(config-if)#x25 wout 7
Router(config-if)#x25 ips 512
Router(config-if)#x25 ops 512
Router(config-if)#^Z
Router#
```

Circuits virtuels

Un routeur CISCO permet d'établir simultanément jusqu'à huit circuits virtuels. On définit le nombre maximum de circuits ouverts simultanément vers le fournisseur avec la sous-commande d'interface `x25 nvc`. Par exemple, pour imposer une limite de deux CV simultanés, on configure l'équipement comme suit :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0
Router(config-if)#x25 nvc 2
Router(config-if)#^Z
Router#
```

Les circuits virtuels sont automatiquement fermés lorsqu'aucune donnée ne les traverse pendant un délai défini par la sous-commande `x25 idle`. Par exemple, pour fermer un circuit au terme de cinq minutes d'inactivité, il faut configurer l'équipement comme suit :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0
Router(config-if)#x25 idle 5
Router(config-if)#^Z
Router#
```

Sur la jonction avec Transpac, le commutateur de ce dernier joue le rôle d'ETCD et le routeur joue le rôle d'ETTD.

On configure les plages de voies logiques, ou circuits virtuels, à l'aide d'un ensemble de sous-commandes de l'interface série. Jusqu'à 4095 voies peuvent être utilisées simultanément. En pratique, on devra n'en utiliser qu'une partie, en fonction de l'abonnement choisi auprès de Transpac.

On distingue quatre plages de voies logiques :

– les circuits virtuels permanents,

- les circuits virtuels entrants, établis par l'ETCD,
- les circuits virtuels sortants, établis par l'ETTD,
- les circuits virtuels bidirectionnels, établis par l'ETTD ou l'ETCD.

Conformément aux indications de Transpac, il faut, au moment de la mise en place du routeur sur l'accès X25, configurer les différentes plages de circuits virtuels afin que notre matériel puisse les choisir correctement sans entraîner de collision.

Le tableau suivant indique les différentes sous-commandes de configuration disponibles, qui doivent être suivies d'un numéro de voie logique :

plage	**commande**
plus haut CV entrant	x25 hic
plus bas CV entrant	x25 lic
plus haut CV sortant	x25 hoc
plus bas CV sortant	x25 loc
plus haut CV bidirectionnel	x25 htc
plus bas CV bidirectionnel	x25 ltc

Configuration du numéro d'appel

On configure le numéro d'appel avec la sous-commande d'interface `x25 map` qui permet d'associer l'adresse IP du routeur du fournisseur à son adresse X121. Par exemple, configurons notre équipement pour qu'il contacte le routeur du fournisseur qui dispose de l'adresse IP `192.168.210.1` dans le réseau d'interconnexion, et de l'adresse X121 175111101 :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0
Router(config-if)#x25 map IP 192.168.210.1 175111101 ACCEPT-REVERSE
Router(config-if)#^Z
Router#
```

Nous avons ici utilisé le paramètre facultatif `ACCEPT-REVERSE` qui autorise notre routeur à accepter les appels du fournisseur, qui sont du type « taxation au demandé ».

Activation du routage

Nous activons le routage RIP pour permettre au routeur de joindre les différents brins Ethernet internes. Il ne faut évidemment pas faire d'annonce sur l'interface série, sous peine de voir un circuit virtuel établi en permanence. On configure donc le routage ainsi :

```
Router(config)#router rip
Router(config-router)#network 192.168.22.0
Router(config-router)#passive-interface serial 0
Router(config-router)#^Z
Router#
```

Configuration complète

L'annexe A.3 page 460 présente le fichier de configuration d'un routeur CISCO 1005 pour un accès X25 sur Transpac.

DEUXIÈME PARTIE

Services de base

Les services de base offerts par l'Internet sont au nombre de trois :

1. le service de noms permet d'attribuer des noms aux machines ;
2. la messagerie permet d'échanger du courrier aux quatre coins du globe, en quelques secondes ;
3. les forums acheminent plus de cent mille articles chaque jour, parmi près de dix mille groupes d'intérêt.

4

Le Service de Noms

Avec le nombre croissant de machines sur l'Internet, un système d'attribution de noms rendant les adresses IP transparentes a été mis en place afin de simplifier la vie des utilisateurs. Un plan d'attribution de noms a été défini et des protocoles adaptés ont été développés. On l'appelle DNS.

4.1 Historique

Le service DNS (Domain Name System) consiste en un annuaire gigantesque et distribué sur la planète. Il a été mis en place à l'origine pour associer des adresses IP à des noms de machine, mais de nombreuses extensions y ont été apportées et il est maintenant capable de fournir de nombreuses autres informations.

De nombreux RFC ont été écrits à propos du DNS. Ce sont les RFC 1034 et 1035 qui en précisent les fondements, autant au niveau des concepts que du protocole.

Le DNS repose sur trois composantes :

- un plan d'attribution de noms,
- un protocole évolutif (basé sur des échanges UDP et TCP),
- des serveurs qui coopèrent sur l'Internet pour fournir un service ininterrompu.

À l'origine, l'association entre les noms de machines et les adresses IP était maintenue par le NIC (Network Information Center) dans le fichier `NETINFO:HOSTS.TXT` de la machine `SRI-NIC.ARPA` d'adresse `10.0.0.51`. Il fallait récupérer régulièrement par FTP ce fichier sur le compte `ANONYMOUS` et avec le mot de passe `GUEST`. Les mises à jour et ajouts

à ce fichier étaient faites par courrier électronique (`NIC@SRI-NIC`) ou bien par contact téléphonique (`(415) 859-4775`).

Évidemment, avec l'accroissement du nombre de machines sur l'Internet, cette procédure est rapidement devenue ingérable. En effet, le débit consommé lors de la distribution d'une nouvelle version de la base par cette méthode est proportionnel au carré du nombre de machines présentes sur le réseau. Même une hiérarchisation de la procédure n'aurait pu contenir le trafic ainsi généré.

4.2 Hiérarchie DNS

4.2.1 Concepts de base

Pour mettre en place le DNS, une hiérarchie de domaines a été érigée. La racine de l'arbre s'appelle Point et s'écrit « `.` ». Les domaines sous Point s'appellent des TLD (Top Level Domains). Par exemple, `fr` est un TLD. Il est destiné à un usage français. Le nombre de TLD est fixe, tandis que des sous-domaines des TLD sont crées à la demande par les différents organismes qui les gèrent. Par exemple, `FeNETre.fr` est un sous-domaine de `fr`. Un nombre quelconque de niveaux d'arborescence peut être créé. La société Fenêtre peut décider de créer un sous-domaine pour sa filiale en Italie, `Italy.FeNETre.fr`, et un autre pour sa filiale Allemande, `Germany.FeNETre.fr`.

Ces domaines accueillent des noms de machines qui forment les nœuds terminaux de l'arbre ainsi constitué. On utilise le terme « nom de domaine » autant pour parler d'un domaine que d'un nœud terminal, le terme anglais correspondant étant « domain name » qui peut signifier « nom de domaine » ou « nom dans le domaine ».

La figure 4.1 page suivante présente les domaines et sous-domaines de la société Fenêtre. On peut y distinguer la machine `GW` dans le sous-domaine `France` du domaine `fenetre.fr`.

Le RFC 952 définit les règles auxquelles les chaînes de caractères utilisées dans cet arbre doivent se conformer :

- les caractères admis sont les lettres de l'alphabet (non accentuées), les chiffres et le caractère tiret (« - ») ;
- chaque nom de nœud terminal, domaine ou sous-domaine, doit commencer par une lettre ;
- il n'y a pas de distinction entre les majuscules et les minuscules ;
- la chaîne comprend au plus 63 caractères.

Le RFC 1123 a libéralisé un certain nombre de pratiques hors normes qui étaient autorisées par nombre d'implémentations du protocole :

- la chaîne peut maintenant commencer par un chiffre ;
- il est fortement conseillé de supporter des noms jusqu'à 256 caractères.

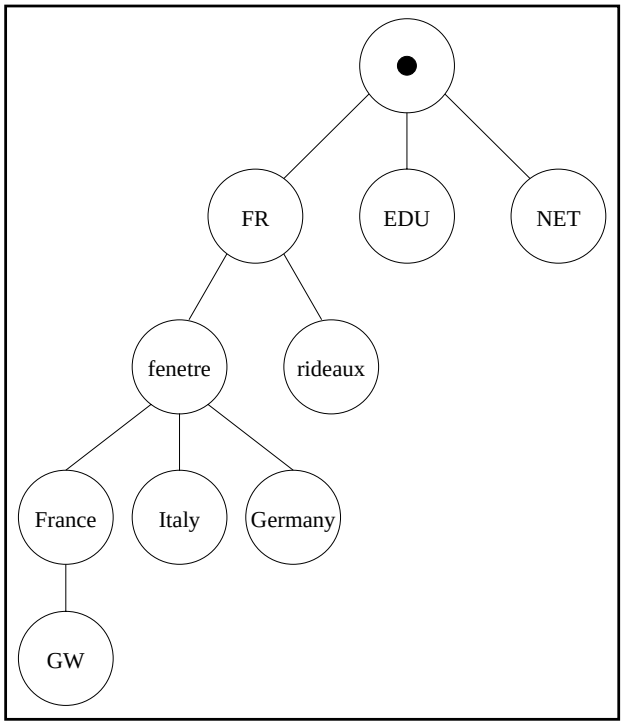

Figure 4.1 *Hiérarchie de domaines*

Un nom de domaine complet doit se terminer par le signe « . » et s'appelle alors un FQDN (Fully Qualified Domain Name). Par exemple, il peut s'agir de « `www.fenetre.fr.` ». Un nom de domaine relatif, par exemple `www` ou `www.fenetre`, est un nom de domaine incomplet qui ne doit alors pas se terminer par « . ». Souvent, on oublie le « . » final des FQDN pour alléger l'écriture mais il est obligatoire dans les fichiers de configuration des serveurs DNS.

4.2.2 Les TLD : Top Level Domain

Les principaux TLD sont décrits dans le tableau 4.1 page suivante. Il en existe trois types :

- les TLD à usage ouvert à tous,
- les TLD réservés pour les États-Unis,
- les TLD représentant les pays.

4.2.3 Le domaine `in-addr.arpa`

Le principe du DNS est d'associer des données à chaque nœud de l'arbre qu'on vient d'étudier. Par exemple, l'adresse `192.168.22.65` va pouvoir être associée au routeur désigné par le nœud terminal `gw` du sous-domaine `France.fenetre.fr`. Mais si on veut pouvoir

Utilisateur	TLD	Usage courant
Pas de restriction	COM	Entités commerciales
	EDU	Universités, écoles diverses
	NET	Fournisseurs Internet
	ORG	Organisations quelconques
	INT	Organisations internationales
USA	GOV	Agences gouvernementales américaines
	MIL	Armée américaine
Pays divers	FR	France
	US	USA
Réseau ARPA	ARPA	Ce TLD accueille in-addr.arpa, pour la gestion des zones inverses

Tableau 4.1 *Les différents TLD*

à l'inverse associer le nom de machine `gw` à l'adresse IP `192.168.22.65`, il faut pouvoir placer cette dernière dans notre arbre.

Pour cela, le domaine `in-addr.arpa` a été réservé. Les adresses IP sont découpées en quatre octets et ces derniers sont considérés comme des noms de domaines. Le principe d'un arbre étant de rapprocher de la racine les éléments les plus importants, on place les octets de poids fort en haut de l'arbre, comme on peut le constater sur la figure 4.2 page suivante.

À partir de ce principe, les serveurs et protocoles en jeu dans le DNS vont pouvoir traiter les adresses IP de la même façon que les noms de domaine. C'est aux administrateurs de vérifier que le nom associé à une adresse IP de l'arbre et que l'adresse IP associée au même nom coïncident.

4.3 Transferts de zones

Une même machine ne peut héberger qu'un serveur DNS pour toutes les opérations liées aux protocoles associés. Ce serveur va interroger ses homologues et répondre à des requêtes. Pour optimiser l'utilisation du débit disponible et les délais, les serveurs possèdent un cache qui mémorise les réponses reçues.

Les domaines et sous-domaines sont découpés en zones. Il en existe deux types :

- les zones directes (*direct zones*) qui permettent principalement d'associer des numéros IP à des noms de machines ;
- les zones inverses (*reverse zones*) qui se situent sous `in-addr.arpa` et qui permettent notamment d'associer des noms de machines à des adresses IP.

Les informations caractérisant une zone, par exemple les adresses IP des noms des machines qui en font partie, sont regroupées dans un fichier unique appelé « fichier de zone ». Différents

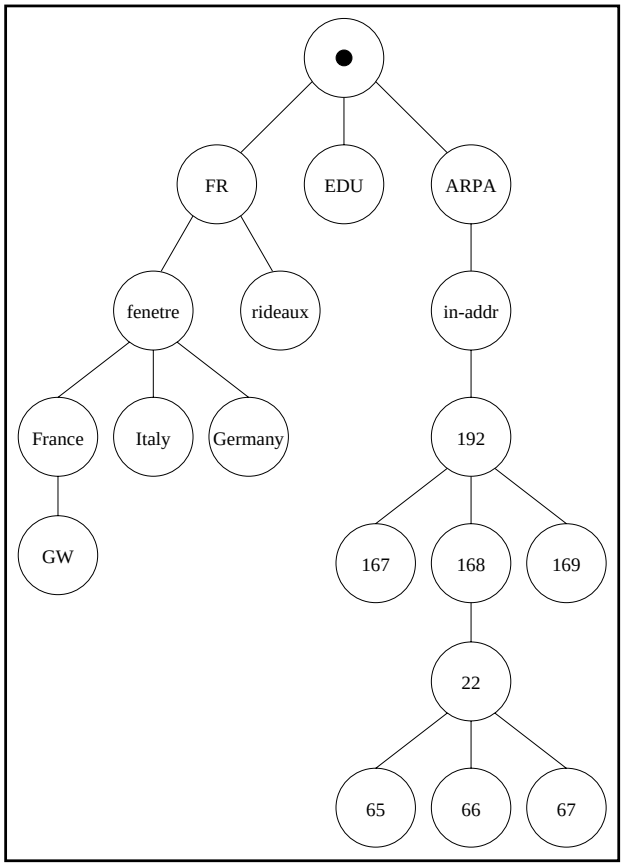

Figure 4.2 *Hiérarchie sous* `in-addr.arpa`

serveurs vont entrer en jeu pour maintenir à jour ce fichier et distribuer son contenu, on les appelle des serveurs qui font autorité (*authoritative servers*) et on dit que les réponses qu'ils fournissent à propos de la zone qu'ils maintiennent font autorité (*authoritative answers*). Cette redondance de serveurs pour une même zone est nécessaire car un réseau connecté à l'Internet est virtuellement injoignable si aucun des serveurs DNS gérant les zones auxquelles ses machines appartiennent n'est accessible.

Le premier modèle de gestion d'une zone, défini dans le RFC 1034, repose sur deux types de serveurs :

- le serveur primaire de la zone (*primary server*), sur lequel le fichier de zone est mis à jour par l'administrateur de la zone ;
- le ou les serveurs secondaires de la zone (*secondary servers*), qui vont régulièrement rapatrier le fichier de zone depuis le serveur primaire.

Le RFC 1996, *addendum* au RFC 1034, définit un modèle plus complexe, qui permet de hiérarchiser la distribution des fichiers de zones. On définit pour cela quatre types de serveurs DNS :

- les serveurs esclaves (*slave servers*), qui font autorité et qui utilisent des transferts de zone pour récupérer le fichier correspondant ;

– les serveurs maîtres (*master servers*), qui font autorité et qui acceptent les requêtes de transfert de zone depuis des serveurs esclaves ;

– le serveur maître primaire (*primary master*), qui fait évidemment autorité et qui est la racine du graphe de dépendance de l'arbre ;

– les serveurs furtifs (*stealth servers*), qui sont identiques aux serveurs esclaves mis à part qu'ils ne sont pas référencés dans la liste des serveurs maintenant la zone, liste d'enregistrements de type NS que nous étudierons par la suite.

Pour ajouter un niveau dans l'arbre, il suffit de mettre en place un serveur esclave, afin qu'il rapatrie le fichier de zone, et maître à la fois pour qu'il puisse fournir ces informations à d'autres esclaves. Bien sûr, il faut éviter les boucles dans le graphe de dépendances.

Les serveurs esclaves, maîtres et furtifs sont les secondaires de la zone.

Ces rôles dépendent d'une zone, un même serveur peut donc avoir différents comportements distincts s'il gère plusieurs zones et être ainsi secondaire de la zone `fenetre.fr` et maître primaire de la zone `rideaux.fr`, par exemple.

La figure 4.3 présente un exemple de graphe de dépendances pour une même zone gérée par cinq serveurs : un primaire et quatre secondaires.

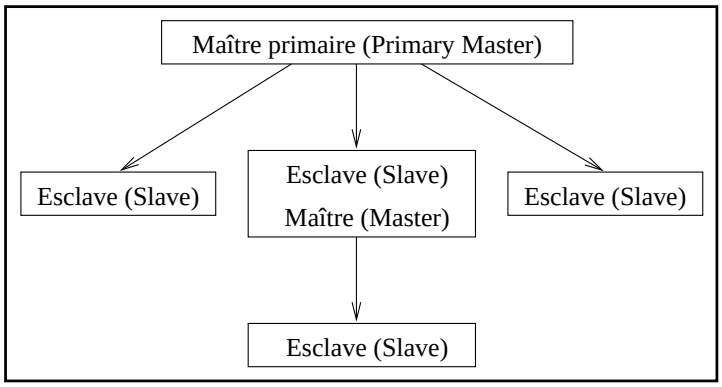

Figure 4.3 *Transferts de zones*

Notons que les serveurs utilisent généralement TCP pour les transferts de zones et UDP pour les autres opérations, même si TCP et UDP peuvent être utilisés indifféremment, à la nuance près que la taille maximale des paquets UDP empêche le transfert de zones importantes.

4.4 Recherche récursive

En plus de ces rôles liés aux zones pour répondre aux requêtes sur les éléments qui y sont contenus, les serveurs DNS savent interroger leurs homologues pour rechercher à travers le labyrinthe des serveurs celui qui contient l'information pertinente.

Par exemple, lorsqu'un navigateur World Wide Web désire afficher le document contenu à l'URL `http://gw.France.fenetre.fr/index.html`, il doit alors se connecter sur la machine `gw.France.fenetre.fr` qui héberge un serveur WWW. Il doit pour cela découvrir son adresse. Il va donc interroger un serveur DNS local s'il en existe un, ou un serveur DNS chez le fournisseur Internet, et charger ce dernier de découvrir l'adresse IP de `gw.France.fenetre.fr`. La requête fournie par le navigateur possède le bit de récursivité, bit indiquant au serveur qu'il doit effectuer une recherche récursive.

Le principe de la recherche récursive est simple : le serveur va tour à tour retransmettre cette requête aux serveurs gérant les sous-domaines correspondant à `gw.France.fenetre.fr`, sans positionner le bit de récursivité. Il va pour cela suivre les étapes que voici, et qu'on peut retrouver sur la figure 4.4 page suivante :

- il interroge son cache pour connaître les adresses IP des serveurs de la racine, c'est-à-dire les serveurs gérant « Point ». Ces serveurs sont indiqués dans un fichier appelé fichier de cache, dont le contenu est chargé dans le cache au démarrage du serveur ;
- il demande l'adresse IP de `gw.France.fenetre.fr` à un serveur de la racine pris au hasard. Ce dernier, qui ne connaît que les données de la zone racine répond par les adresses IP des serveurs gérant `fr` ;
- il interroge un des serveurs gérant `fr`. Celui-ci répond par les adresses IP des serveurs gérant la zone `fenetre.fr` ;
- il interroge un des serveurs gérant `fenetre.fr`. Celui-ci répond par les adresses IP des serveurs gérant la zone `France.fenetre.fr` ;
- il interroge un des serveurs gérant `France.fenetre.fr`. Celui-ci renvoie l'adresse IP de `gw.France.fenetre.fr` ;
- notre serveur renvoie enfin au navigateur la valeur correspondante.

4.5 Serveurs de type *forwarder*

À un serveur DNS susceptible de recevoir des requêtes récursives, on peut associer une liste de *forwarders*. Il s'agit d'un ou plusieurs serveurs auxquels il devra systématiquement retransmettre les requêtes, sans lui-même faire de recherche récursive. Si les *forwarders* ne fournissent pas de réponse satisfaisante, le serveur initial va alors entamer une phase de récursivité, comme on peut le constater sur la figure 4.5 page suivante.

Ce type de comportement est parfois utilisé lors d'un raccordement à faible débit avec le fournisseur Internet. On met alors en place un serveur DNS local. S'il ne possède pas de clause *forwarders* et si une des machines de ce site désire interroger le serveur WWW `gw.France.fenetre.fr`, quatre requêtes, donc huit paquets UDP, vont traverser la ligne de transmission bas débit, souvent accompagnée d'une latence importante. Si on positionne un *forwarder* chez le fournisseur, seuls deux paquets vont traverser la ligne de transmission. On divise ainsi par quatre le temps de latence.

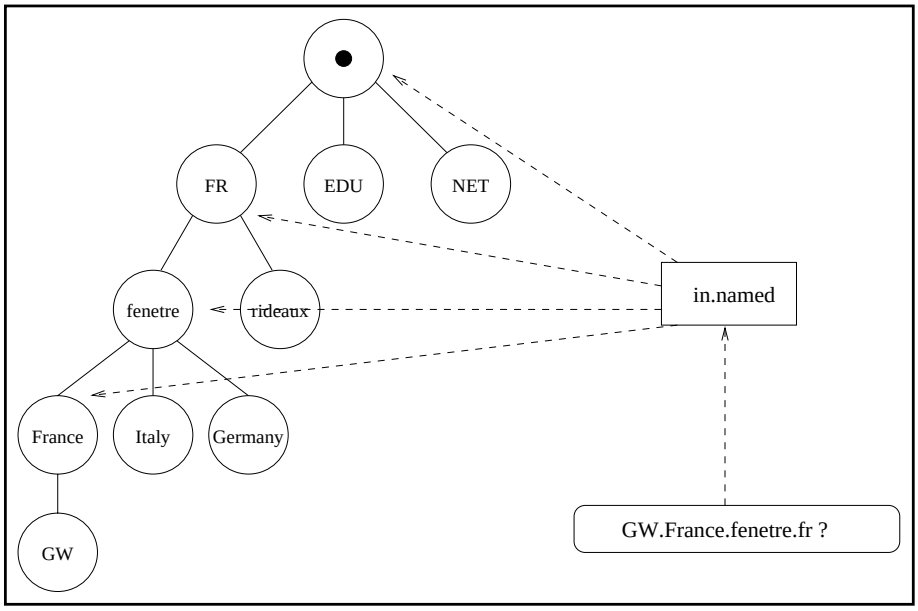

Figure 4.4 *Recherche récursive*

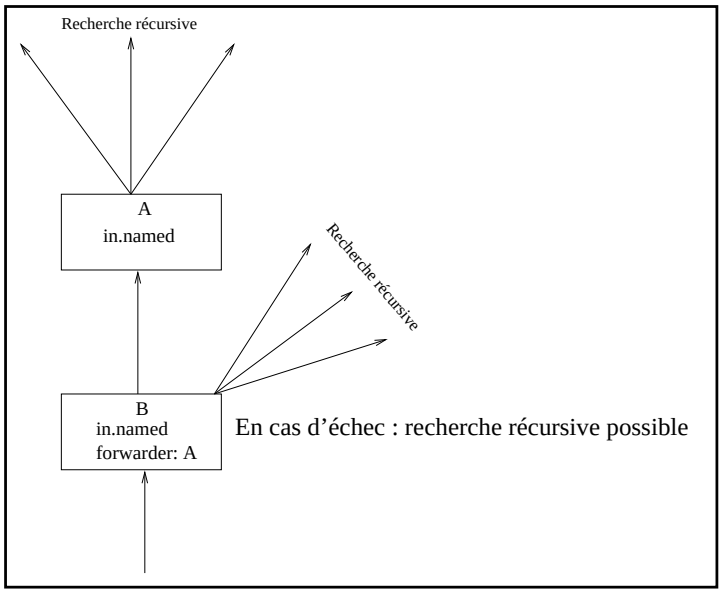

Figure 4.5 *Serveur de type forwarder*

On a dit qu'un serveur DNS muni d'une liste de *forwarders* est susceptible de faire une recherche récursive en dernier recours. On peut empêcher ce comportement en plaçant le serveur en mode esclave, comme on peut le constater sur la figure 4.6 page suivante.

Notons bien que le comportement lié aux *forwarders* et au mode esclave est indépendant des rôles éventuellement joués par le serveur vis-à-vis de zones, tels que les rôles de serveur primaire ou secondaire.

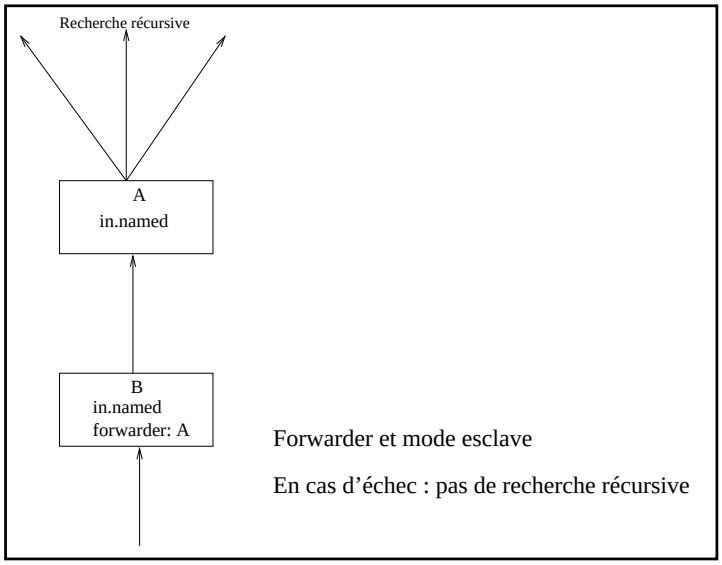

Figure 4.6 *Forwarders et mode esclave*

4.6 Configuration des clients DNS

Un client DNS, que ce soit un poste de travail sous Windows ou une machine Unix, a besoin de connaître au moins un serveur DNS local ou chez le fournisseur afin de résoudre les noms de machines en adresses IP et réciproquement. De plus, lorsque l'utilisateur désigne une machine de son propre domaine, il ne fournit pas à son application un FQDN mais un nom relatif à son propre domaine. Par exemple, sur la machine `PC1.fenetre.fr`, un utilisateur peut vouloir se connecter au serveur WWW de FeNETre et spécifier pour cela `www` plutôt que `www.fenetre.fr`. Il faut donc indiquer à chaque poste client le nom de domaine (`fenetre.fr` dans notre exemple) permettant éventuellement de compléter les noms de machines.

Examinons comment mettre en place ces deux informations sur différents types de systèmes d'exploitation.

4.6.1 Macintosh

Sur Macintosh, il faut indiquer l'adresse IP d'un serveur DNS et le domaine local dans MacTCP, comme on peut le constater sur la figure 2.18 page 57 dans laquelle le nom de domaine est `societe.fr` et l'adresse IP est `192.168.22.35`.

4.6.2 PC - Chameleon

Avec Chameleon sur PC, il faut indiquer le nom du poste, le nom de domaine et l'adresse d'un serveur DNS acceptant les requêtes récursives avec l'onglet ***[Setup/Configuration/IP***

Configuration], comme on peut l'observer sur la figure 4.7.

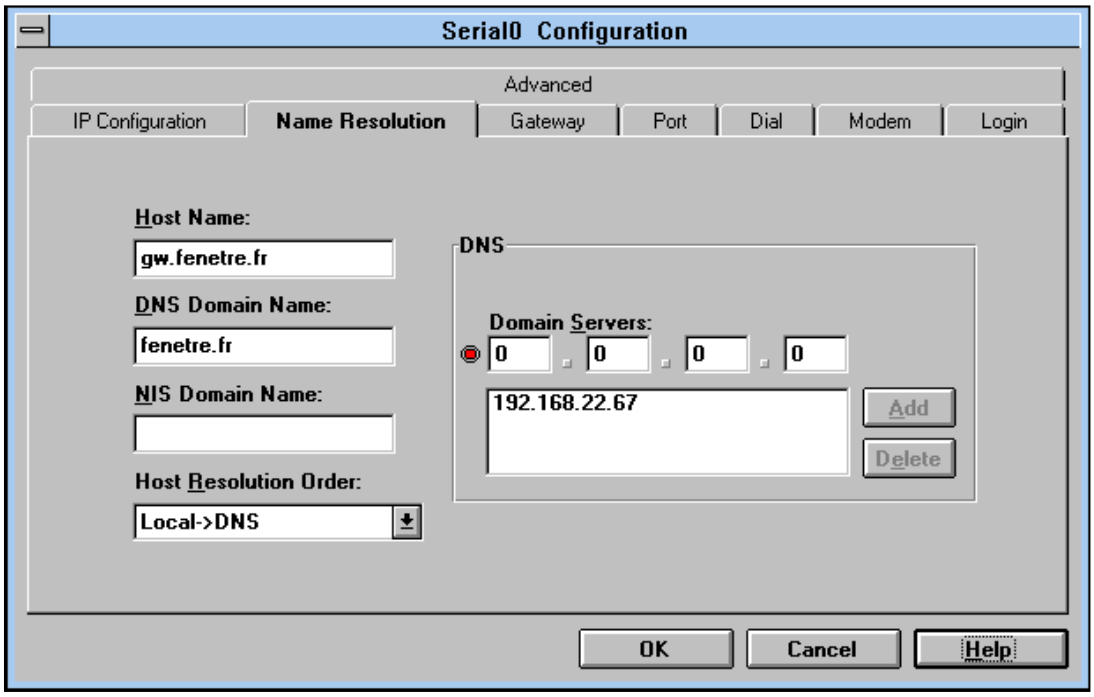

Figure 4.7 *Configuration du DNS avec Chameleon*

4.6.3 Unix

Unix utilise déjà un système de transmission d'informations de ce type à travers le réseau local, il s'agit des NIS (Network Information Services). Il faut donc ajouter les informations provenant du DNS dans ce processus de diffusion d'informations locales.

Deux fichiers de configuration doivent ainsi être adaptés :

- `/etc/resolv.conf`
- `/etc/nsswitch.conf`

Le fichier `/etc/resolv.conf` indique le domaine local par une primitive `domain` et les adresses IP des serveurs de noms par des primitives `nameserver` :

```
<ls@gw.France.fenetre.fr> cat /etc/resolv.conf
domain France.fenetre.fr
nameserver 192.168.22.35
<ls@gw.France.fenetre.fr>
```

Le fichier `/etc/nsswitch.conf` indique la politique d'interrogation des NIS, du DNS et du fichier `/etc/hosts`. On indique sur la ligne commençant par `hosts:` l'ordre des interrogations. Par exemple, pour interroger les NIS, puis le DNS et enfin le contenu du fichier

`/etc/hosts`, on utilise la ligne de configuration suivante :

```
hosts:   nis [NOTFOUND=continue] dns [NOTFOUND=continue] files
```

On peut aussi demander au serveur NIS d'interroger lui-même le démon `in.named` qui gère le service DNS. Pour cela, on passe l'option `-B` au démon NIS `rpc.nisd` et on se contente de la ligne de configuration suivante sur les postes client :

```
hosts:   nis [NOTFOUND=continue] files
```

C'est la bibliothèque `libresol.{so,a}`, avec laquelle une édition de liens statique ou dynamique est réalisée avec les logiciels réseau, qui se charge de lire les fichiers de configuration `/etc/nsswitch.conf` et `/etc/resolv.conf` *à chaque interrogation*, et d'effectuer les requêtes, comme on peut le constater sur la figure 4.8. Il n'y a donc aucun démon à relancer ni à redémarrer la machine quand on apporte une modification à l'un de ces fichiers.

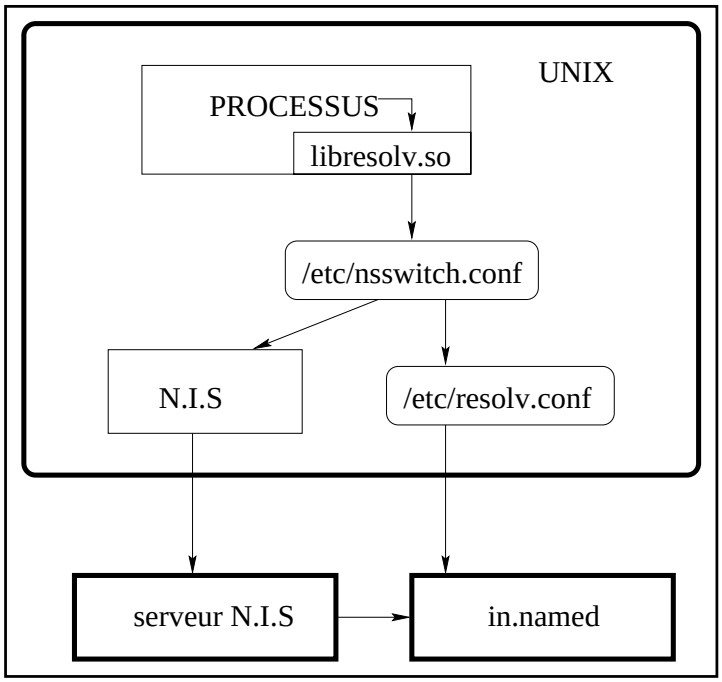

Figure 4.8 *Flux de données entre un client et un serveur DNS*

4.7 Zones et domaines

Distinguons précisément les domaines des zones dont nous avons jusqu'à maintenant parlé.

Un domaine est un nœud de l'arbre d'attribution de noms. Une zone est une entité administrative qui peut englober un ou plusieurs domaines. À chaque zone est associé un fichier de zone, et un certain nombre de serveurs qui coopèrent pour distribuer les informations qui y sont incluses.

Reprenons notre exemple de la société fenêtre et de ses multiples filiales, dont notamment celles établies en France et en Italie.

Les différents domaines en jeu sont alors `fenetre.fr`, `France.fenetre.fr` ainsi que `Italy.fenetre.fr`.

On peut les rassembler au sein d'une même zone `fenetre.fr` ou de trois zones distinctes, `France.fenetre.fr`, `Italy.Fenetre.fr` et `fenetre.fr`, comme on peut le constater sur la figure 4.9.

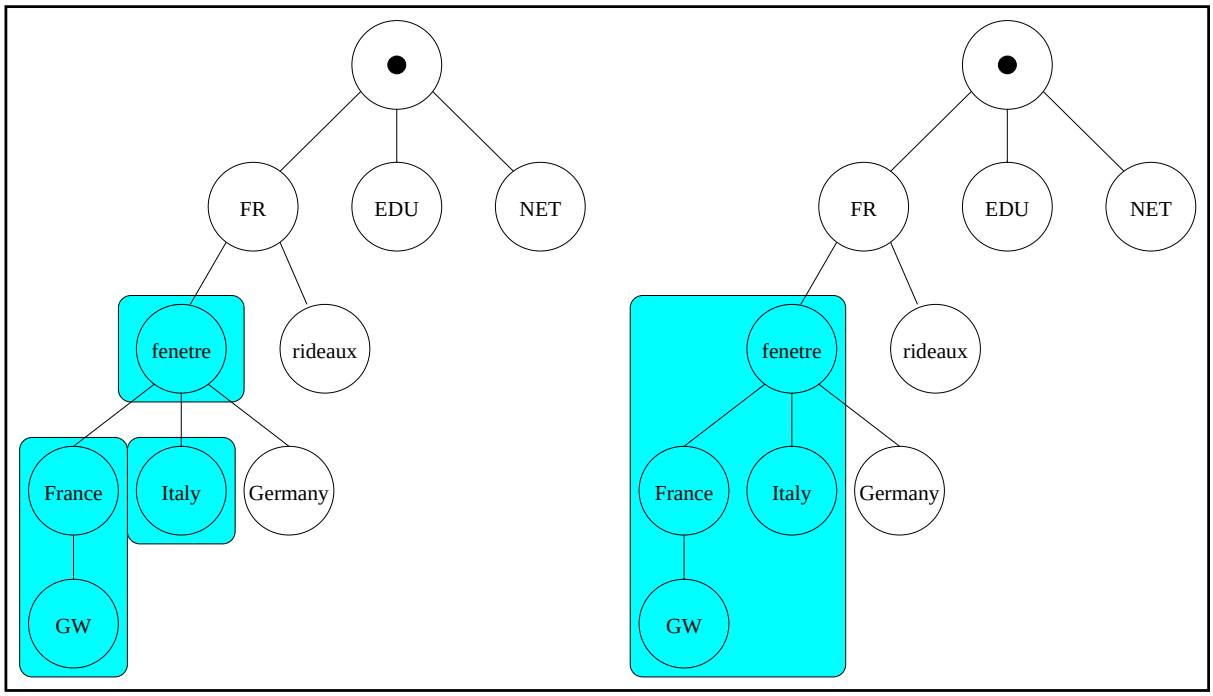

Figure 4.9 *Différents découpages de domaines en zones*

On pourrait aussi imaginer créer une zone `fenetre.fr` contenant `fenetre.fr` ainsi que `France.fenetre.fr`, et une autre zone contenant uniquement `Italy.fenetre.fr`.

4.8 Types d'enregistrements

On a dit jusqu'ici qu'on peut associer une adresse IP à un nœud de l'arbre, ou un nom de domaine s'il s'agit d'un nœud terminal situé sous `in-addr.arpa`.

Il existe de nombreux autres enregistrements possibles, qu'on appelle RR (*resource records*).

Une syntaxe particulière est associée à chacun d'eux, syntaxe qui comprend trois parties :

- La partie gauche de l'enregistrement indique son **possesseur**. Il s'agit du nœud de l'arbre auquel s'applique cet enregistrement.
- La partie centrale indique une durée de vie, la classe et le type d'enregistrement. La durée de vie (TTL, Time To Live) est celle de l'information dans les caches. Elle est souvent omise car l'enregistrement SOA en fournit une valeur par défaut. La classe identifie la famille de protocoles, il s'agit de IN pour l'Internet, CH pour le réseau Chaos, etc. Par la suite, on utilisera uniquement la classe IN.
- La partie droite indique la valeur de l'enregistrement. Elle peut comporter plusieurs champs.

4.8.1 Enregistrements de type SOA

Le SOA est l'enregistrement le plus important d'une zone. Il est unique et comprend de nombreux champs.

En voici un exemple :

```
@               IN SOA          ns.fenetre.fr. ls.fenetre.fr. (
                                1996120101      ;Serial
                                28800           ;Refresh (8 heures)
                                7200            ;Retry   (2 heures)
                                604800          ;Expire  (7 jours)
                                86400 )         ;Minimum (1 jour)
```

Voici la signification de ses champs (les mesures de temps sont exprimées en secondes) :

- le caractère « @ » indique que le possesseur de l'enregistrement est la zone elle-même ;
- le champ IN SOA indique que cet enregistrement est du type SOA dans le domaine Internet ;
- le champ `ns.fenetre.fr` est le nom du maître primaire ;
- le champ `ls.fenetre.fr` indique que le responsable de la zone possède l'adresse électronique `ls@fenetre.fr` ;
- le champ `Serial` indique un numéro de série, souvent fabriqué par l'administrateur système à partir de la date à laquelle il fait les modifications. Quand un secondaire désire rapatrier un fichier de zone, il consulte l'enregistrement SOA du primaire, et si le champ `Serial` a été incrémenté, il rapatrie la zone par une requête de type AXFR ; certains serveurs utilisent des requêtes de type IXFR pour effectuer un rapatriement incrémental de la zone, comme décrit dans le RFC 1996 ;
- le champ `Refresh` indique la fréquence à laquelle un secondaire examine le SOA du primaire pour savoir si un transfert de zone est nécessaire ;
- le champ `Retry` indique la fréquence à laquelle un secondaire interroge un primaire après un premier échec ;

- le champ `Expire` indique le délai au terme duquel un serveur secondaire décide d'invalider une zone dont le primaire est inaccessible ;
- le champ `Minimum` indique la durée de vie par défaut associée aux différents enregistrements de la zone.

Le tableau suivant indique les valeurs recommandées pour ces paramètres par le RFC 1537 :

Type de zone	TLD	Domaine quelconque
Refresh	24 heures	8 heures
Retry	2 heures	2 heures
Expire	30 jours	7 jours
Minimum TTL	4 jours	1 jour

4.8.2 Enregistrement de type A

L'enregistrement de type A permet d'associer une adresse IP à un nom de machine ou de domaine.

Par exemple, si le routeur `gw.fenetre.fr` possède une interface de type Ethernet et d'adresse `192.168.22.35`, ainsi qu'une interface PPP d'adresse `192.168.200.2`, on lui associe les deux enregistrements A qui suivent, dans la zone `fenetre.fr` :

```
gw              IN A            192.168.22.35
gw-ppp          IN A            192.168.200.2
```

4.8.3 Enregistrement de type PTR

Cet enregistrement se trouve le plus souvent dans les zones inverses et associe alors un nom de machine à un numéro IP. Par exemple, la zone inverse `200.168.192.in-addr.arpa` contient :

```
35              IN PTR          gw.fenetre.fr.
```

4.8.4 Enregistrement de type CNAME

Ce type d'enregistrement permet de définir des alias appelés CNAME (Common Name).

On utilise souvent ce type d'enregistrement pour créer le nom d'un serveur WWW. Par exemple, supposons que le serveur WWW de la société Fenêtre soit hébergé sur la machine `gw.fenetre.fr`. Il suffit pour déclarer le serveur WWW de créer dans la zone

`fenetre.fr` l'enregistrement suivant :

```
www             IN CNAME        gw.fenetre.fr.
```

C'est plus souple que de changer le nom `gw.fenetre.fr` en `www.fenetre.fr`, notamment si on est amené à migrer le serveur WWW à l'avenir.

Une règle importante indique que le nom dans la partie droite d'un CNAME ne doit pas être à gauche d'un CNAME dans un autre enregistrement. Il faut donc s'abstenir de chaîner les CNAME.

4.8.5 Enregistrement de type NS et mise en place d'une délégation

Ce type d'enregistrement permet de préciser la liste des serveurs DNS. On l'utilise pour donner la liste des serveurs primaire et/ou secondaires de la zone courante, ainsi que pour déléguer des sous-domaines à d'autres serveurs.

Par exemple, la zone `fenetre.fr`, qui possède un serveur primaire sur le réseau de la société fenêtre et un serveur secondaire chez le fournisseur Internet, contient les enregistrements suivants :

```
@               IN NS           ns.fenetre.fr.
                IN NS           ns.fournisseur.fr.
```

On remarque ici que la partie gauche d'un enregistrement n'a pas besoin d'être répétée. Par défaut, c'est la dernière partie gauche présente qui s'applique.

`fenetre.fr` possède un sous-domaine `Italy.fenetre.fr` qui est géré directement par la filiale italienne et par son fournisseur sur les machines `ns.Italy.fenetre.fr` et `ns.fournisseur-italien.it`. La zone `fenetre.fr` contient donc aussi :

```
Italy           IN NS           ns.Italy.fenetre.fr.
                IN NS           ns.fournisseur-italien.it.
```

Ajouter ce type d'enregistrement s'appelle **mettre en place une délégation**. Le domaine `Italy.fenetre.fr` est ici délégué à la filiale italienne.

Il manque néanmoins une information nécessaire dans la zone `fenetre.fr`, suite à cette délégation : il s'agit de l'adresse IP de `ns.Italy.fenetre.fr`. En effet, lors d'une recherche récursive, un serveur DNS qui désire connaître par exemple l'adresse IP de la machine `lasagnes.Italy.fenetre.fr` va interroger un des serveurs faisant autorité pour `Italy.fenetre.fr` : `ns.fournisseur.fr` ou `ns.fenetre.fr`. Celui-ci va renvoyer les noms des serveurs qui font autorité pour la zone `Italy.fenetre.fr`. Le DNS qui fait cette recherche récursive va donc alors en choisir un au hasard, afin de l'interroger ; il

pourra donc s'agir de `ns.Italy.fenetre.fr`. Mais avant de pouvoir l'interroger, il faut connaître son adresse IP, c'est-à-dire faire une recherche récursive, et on tombe alors dans une boucle, car `ns.Italy.fenetre.fr` est dans le domaine de `Italy.fenetre.fr` tout comme `lasagnes.Italy.fenetre.fr`.

La solution consiste à configurer `ns.fenetre.fr` convenablement pour qu'il indique l'adresse IP du serveur `ns.Italy.fenetre.fr` lorsqu'on l'interroge à propos de la machine `lasagnes.Italy.fenetre.fr`.

On doit donc ajouter à la zone `fenetre.fr` un enregistrement de type A qu'on appelle **glue** :

```
ns.Italy.fenetre.fr.   IN A      192.168.210.15
```

ATTENTION

Une erreur trop souvent commise par les administrateurs de DNS est d'ajouter des enregistrements de glue dans des cas de figure où cela n'est pas nécessaire.
Cette redondance d'informations peut créer de graves problèmes lors de modifications ultérieures des adresses IP correspondantes. Par exemple, il ne faut pas ajouter l'adresse IP du serveur du fournisseur italien :

```
; enregistrement inutile et nuisible :
ns.fournisseur-italien.it.  IN A    192.168.211.2
```

En effet, si le fournisseur italien renumérote ses machines, la zone `fenetre.fr` devient alors incorrecte.

4.8.6 Enregistrement de type MX

Acheminement du courrier

Les enregistrements de type MX permettent d'acheminer le courrier en indiquant des échangeurs de *mail*.

Ils fournissent un nom de passerelle SMTP ainsi qu'un poids qui permet de classer les réponses.

Par exemple, la société Fenêtre dispose d'un serveur SMTP de nom `mail.fenetre.fr`. La zone `fenetre.fr` contient donc l'enregistrement suivant :

```
@              IN MX      10   mail.fenetre.fr.
```

Lorsqu'un courrier destiné à un nom de machine ou de domaine particulier est fourni à une

passerelle SMTP, celle-ci utilise le DNS pour déterminer à quelle autre passerelle le transmettre.

Prenons pour exemple l'adresse *email* suivante : `ls@host.fenetre.fr`.

La passerelle effectue les étapes suivantes dans l'ordre, jusqu'à obtenir une réponse favorable du DNS :

- recherche d'enregistrements de type MX pour `host.fenetre.fr`,
- recherche d'enregistrements de type A pour `host.fenetre.fr`.

S'il existe des enregistrements de type MX, elle en choisit un parmi ceux de poids le plus faible, et livre le courrier à la machine portant le nom correspondant.

Sinon, s'il existe des enregistrements de type A, elle en choisit un et livre le message à la machine portant l'adresse correspondante.

Si aucun enregistrement n'existe, la passerelle SMTP met le courrier de côté et essaie à nouveau par la suite. Au bout d'un certain nombre d'essais infructueux, elle envoie un message pour avertir son auteur du délai, et encore un peu plus tard elle efface le courrier en prévenant à nouveau son auteur.

Traitement particulier des enregistrements de type MX

Le DNS possède des comportements particuliers quant aux MX.

Tout d'abord, le DNS associe à un nom possédant un CNAME les MX de la partie droite du CNAME. Ainsi, le courrier envoyé à destination de XYZ est livré aux échangeurs de *mail* d'UVW si XYZ est un CNAME pour UVW.

D'autre part, on peut configurer le DNS avec des enregistrements d'un type particulier, appelés *wildcard MX records*. Un MX de ce type correspond pour la zone `fenetre.fr` à un enregistrement de cette forme :

```
*               IN MX     100   mail.fenetre.fr.
```

Il permet d'indiquer au DNS de générer un MX par défaut de poids 100 à destination de `mail.fenetre.fr` pour toute interrogation de type MX sur un élément faisant partie de la zone `fenetre.fr` et *ne comportant aucun autre enregistrement*.

Ainsi, un courrier à destination de `user@ca-n-existe-pas.fenetre.fr` va être acheminé vers `mail.fenetre.fr`.

Rappelons-nous maintenant que la zone `fenetre.fr` contient l'enregistrement :

```
gw              IN A            192.168.22.35
```

Un courrier à destination de `user@gw.fenetre.fr` ne sera donc pas livré au serveur SMTP `mail.fenetre.fr` mais à `gw.fenetre.fr` car aucun MX, même de type *wildcard*, n'est rattaché à `gw`. En effet ce nom possède déjà au moins un enregistrement, le MX *wildcard* ne s'applique donc pas.

Mise en place des enregistrements de type MX

Deux politiques d'acheminement des courriers sont généralement utilisées, en fonction du type de raccordement avec le fournisseur : raccordement permanent ou intermittent.

Dans le cadre d'un raccordement permanent à l'Internet, on met en place les MX suivants :

- un MX de poids faible pour le domaine, vers la passerelle SMTP interne, pour acheminer les messages à destination d'adresses du type `user@fenetre.fr` ;
- un MX de poids plus important pour le domaine, vers une passerelle SMTP chez le fournisseur pour acheminer correctement les messages à destination d'adresses du type `user@fenetre.fr`, lorsque la passerelle interne est en panne ;
- un MX *wildcard* de poids faible pour le domaine, vers la passerelle SMTP interne, pour acheminer les messages à destination d'adresses du type `user@xyz.fenetre.fr` où xyz n'existe pas ;
- un MX *wildcard* de poids important pour le domaine, vers une passerelle SMTP chez le fournisseur, pour acheminer correctement les messages à destination d'adresses du type `user@xyz.fenetre.fr` où xyz n'existe pas, lorsque la passerelle interne est en panne ;
- un MX de poids faible, pour chacun des noms contenus dans la zone, vers la passerelle SMTP interne ;
- un MX de poids plus important, pour chacun des noms contenus dans la zone, vers une passerelle SMTP chez le fournisseur.

Ainsi, la zone `fenetre.fr` contient les MX suivants :

```
@               IN MX      10   mail.fenetre.fr.
                IN MX     100   mail.fournisseur.fr.
*               IN MX      10   mail.fenetre.fr.
                IN MX     100   mail.fournisseur.fr.
gw              IN MX      10   mail.fenetre.fr.
                IN MX     100   mail.fournisseur.fr.
```

La figure 4.10 page ci-contre présente un exemple d'acheminement d'un courrier dans le cadre d'une connexion permanente à l'Internet.

Si le réseau local est connecté au fournisseur de façon intermittente, il ne faut pas faire pointer de MX vers la passerelle SMTP locale car elle n'est pas connectée en permanence et de nombreux messages seraient ainsi refusés.

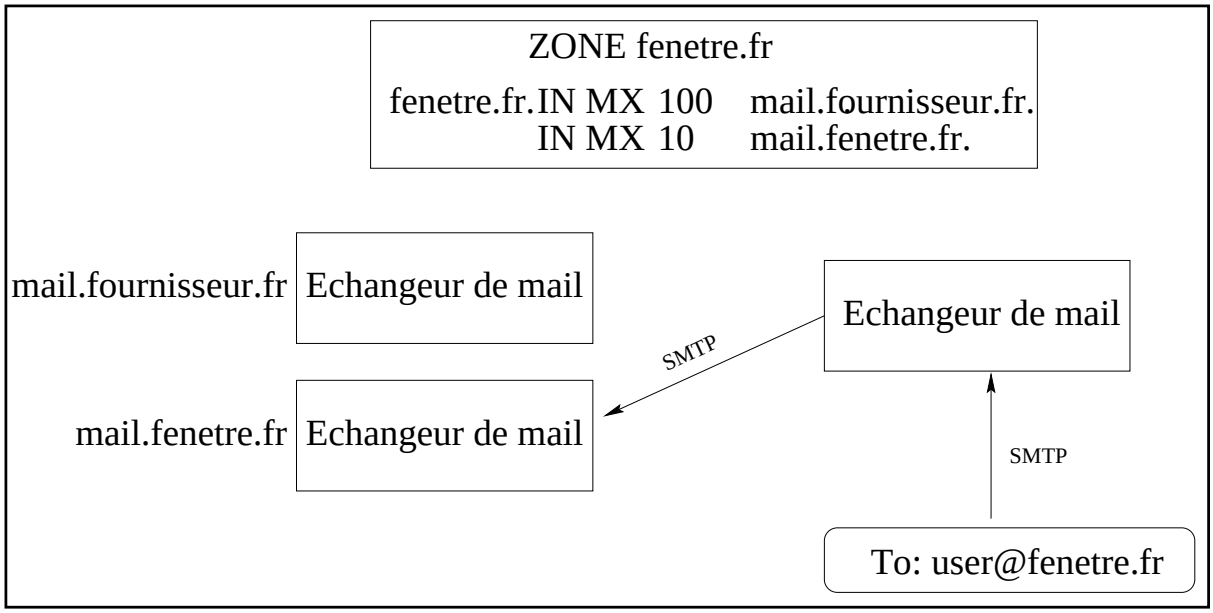

Figure 4.10 *Echangeurs de mail*

On dirige alors les MX uniquement vers le fournisseur, à charge pour ce dernier de mettre en place une procédure automatique pour envoyer les *mails* stockés chez lui lorsque le réseau local se retrouve connecté.

Indication

Il est fortement conseillé de ne pas mettre de champ MX pour des sous-domaines dans le fichier de zone du domaine en question.
C'est-à-dire que dans le fichier de zone de `fenetre.fr`, on ne doit pas enregistrer de MX pour `Italy.fenetre.fr`.
Les MX pour ce dernier domaine doivent être positionnés dans la zone `Italy.fenetre.fr` elle-même. En effet, l'administrateur d'un domaine n'a pas à contrôler l'acheminement des messages à destination des sous-domaines dont il a délégué la gestion.

4.8.7 Enregistrement de type TXT

Ce type d'enregistrement permet d'associer un texte quelconque à son possesseur. Par exemple :

```
gw              IN TXT          "Machine appartenant a Luc Stoned"
```

4.8.8 Enregistrement de type HINFO

Ce type d'enregistrement indique la nature de la carte processeur et le système d'exploitation de la machine visée.

Par exemple :

```
gw              IN HINFO         SUN-IPC Solaris-2.5
```

4.8.9 Enregistrement de type GPOS

Ce type d'enregistrement expérimental, décrit dans le RFC 1712, permet de localiser une machine de façon géographique. Il en indique les latitude (degrés), longitude (degrés) et altitude (mètres).

Par exemple :

```
gw              IN GPOS          -32.6882 116.8652 10.0
```

4.8.10 Enregistrement de type X25

Ce type d'enregistrement expérimental, décrit dans le RFC 1183, permet d'associer une adresse X.121 à une machine.

Par exemple :

```
gw              IN X25           311061700956
```

4.8.11 Enregistrement de type ISDN

Ce type d'enregistrement expérimental, décrit au sein du RFC 1183, permet d'associer un numéro de téléphone RNIS à une machine. Une sous-adresse peut être éventuellement fournie.

Par exemple, le numéro 150862028003217 et la sous-adresse 004 sont associés à la machine `gw` :

```
gw              IN ISDN          150862028003217 004
```

4.8.12 Autres types d'enregistrements

De nombreux autres types d'enregistrements ont été définis. Ils sont pour la plupart considérés comme expérimentaux et sont rarement implémentés ni utilisés. Il s'agit notamment de :

- WKS (Well Known Services) : indication des services fournis par une machine ;
- RP (Responsible Person) : personne responsable de l'équipement ou du domaine ;
- NULL : enregistrement sans contenu.

4.8.13 Nom de réseau, masque de sous-réseaux, adresse réseau, nom d'organisation

Il est possible d'utiliser le DNS pour indiquer des noms de réseaux, des masques de sous-réseaux, des adresses de réseaux et associer au nom d'une organisation l'ensemble des réseaux dont elle a obtenu la délégation. On utilise pour cela les enregistrements de type A et PTR de manière inhabituelle, comme le décrit le RFC 1101.

Nom de réseau

Pour associer un nom de réseau à une adresse réseau, on utilise un enregistrement de type A. Par exemple, pour donner le nom `ls-net.fenetre.fr` au réseau de classe C d'adresse `192.168.22.0`, on ajoute dans la zone `22.168.192.in-addr.arpa` l'enregistrement suivant :

```
0.22.168.192.in-addr.arpa.    IN PTR    ls-net.fenetre.fr.
```

Masque de sous-réseaux

Ce réseau possède un masque de sous-réseaux de valeur `255.255.255.224`. On l'indique avec un enregistrement de type A, toujours dans la zone inverse :

```
0.22.168.192.in-addr.arpa.    IN A      255.255.255.224
```

Adresse réseau

Pour associer une adresse réseau à un nom de réseau, il faut utiliser un enregistrement de type PTR dans la zone directe à laquelle appartient le nom de réseau en question. Par exemple, au nom de réseau `ls-net.fenetre.fr` correspond le réseau de classe C `192.168.22.0`.

On définit pour cela l'enregistrement suivant :

```
ls-net.fenetre.fr.            IN PTR    0.22.168.192.in-addr.arpa.
```

Nom d'organisation

Pour associer différents réseaux au nom de l'organisation à laquelle ils ont été délégués, on utilise des enregistrements de type PTR dont le possesseur est le nom du domaine de l'organisation. Par exemple, si la société Fenêtre utilise le réseau de classe C `192.168.22.0` et le réseau de classe B `172.16.0.0`, elle définit l'enregistrement suivant dans la zone `fenetre.fr` :

```
fenetre.fr.                   IN PTR    0.22.168.192.in-addr.arpa.
                              IN PTR    0.0.16.172.in-addr.arpa.
```

4.9 Le serveur BIND

Le serveur `BIND` (Berkeley Internet Name Domain Server) de l'université de Berkeley, dont une partie importante a été écrite par Paul VIXIE, constitue la référence parmi les serveurs DNS.

4.9.1 Contenu de la distribution

La plupart des Unix commerciaux sont fournis avec un serveur DNS. Il s'agit très souvent de l'adaptation d'une version de `BIND`. L'utilisateur a le choix d'utiliser la version fournie avec le système ou de récupérer la dernière mise à jour de ce logiciel qu'on peut utiliser librement.

Aujourd'hui, on trouve la version stable 4.9.4 par exemple à l'URL suivant :
`ftp://ftp.vix.com/pub/bind/release/4.9.4/bind-4.9.4-REL.tar.gz`

Une pré-version 4.9.5 est aussi disponible à l'URL suivant :
`ftp://ftp.vix.com/pub/bind/testing/bind-4.9.5-T3B.tar.gz`

`BIND` consiste en un démon `named`[1] et une bibliothèque nommée `resolver`.

Le démon permet de servir les requêtes DNS et la bibliothèque est utilisée pour compiler des applications qui ont besoin de faire appel aux services DNS en tant que clients.

`BIND` peut être utilisé pour gérer un nombre quelconque de zones en jouant le rôle de primaire ou de secondaire. Il peut aussi jouer le rôle de *forwarder* pour retransmettre des requêtes.

1. `named` est aussi parfois nommé `in.named`

Dans tous les cas, il utilise un cache en mémoire pour réduire le nombre de requêtes, le débit utilisé et les temps de latence. Lorsqu'il ne fait autorité pour aucune zone, on dit alors qu'il est de type « *cache only* ». On parle ainsi de « *caching only server* ».

La distribution de `BIND` contient différents programmes :

- on trouve un ensemble de programmes permettant le bon fonctionnement et l'administration du service de noms :
  - `named` : il s'agit du démon ;
  - `named-xfer` : cet utilitaire appelé directement par `named` permet d'effectuer les transferts de zones ;
  - `named.reload` : cette commande permet de forcer `named` à relire les fichiers de configuration et à recharger les fichiers des zones pour lesquelles il est secondaire, si les *serial* ont changé. Il utilise pour cela le signal `SIGHUP` ;
  - `named.restart` : cette commande permet de détruire le démon `named` par l'émission d'un signal `SIGKILL` et d'en redémarrer un nouveau ;
  - `ndc` : cet utilitaire permet d'envoyer des signaux divers à `named` afin de lui faire effectuer différentes opérations. Les commandes permises sont les suivantes :
    - `status` : affiche l'état du serveur ;
    - `dumpdb` : force `named` à écrire le contenu de sa base de données et de son cache dans `/var/tmp/named_dump.db` ;
    - `reload` : équivalent de `named.reload` ;
    - `status` : force `named` à remplir `/var/tmp/named.stats` avec des statistiques ;
    - `trace` : incrémente le niveau de *debug* d'une unité ; les informations de *debug* sont inscrites dans `/var/tmp/named.run` ;
    - `notrace` : arrête l'inscription d'informations de *debug* ;
    - `querylog` : force `named` à indiquer par la fonctionnalité `syslog` les différentes requêtes qu'il reçoit ; cette option consomme énormément de ressources sur le système de fichiers ;
    - `start` : active le démon si ce n'est déjà fait ;
    - `stop` : détruit le démon s'il est actif ;
    - `restart` : équivalent de `named.restart` ;
- on trouve aussi un ensemble d'utilitaires permettant d'interroger les serveurs DNS en émettant des requêtes de tous types :
  - `nslookup` : cet utilitaire permet de générer des requêtes DNS afin, par exemple, de tester un serveur ;
  - `dig` : offre sensiblement les mêmes fonctionnalités que `nslookup` mais sa syntaxe est plus compacte, tous les paramètres étant passés sur la ligne de commande ;
  - `dnsquery` : autre utilitaire d'interrogation d'un DNS ;
  - `host` : cette commande permet d'obtenir la ou les adresses IP correspondant à un nom de machine, en interrogeant un DNS.

4.9.2 Fichiers de configuration de BIND

Le démon `BIND` utilise trois types de fichiers de configuration :

- un fichier désigné par « fichier de démarrage » et nommé `/etc/named.boot`. Celui-ci permet de définir différentes options et de localiser le répertoire où se trouvent les autres fichiers de configuration ;
- un ou plusieurs fichiers de cache, dont le contenu est chargé dans le cache du démon au démarrage ; ce fichier est habituellement nommé `named.cache` ;
- différents fichiers de zones.

La figure 4.11 présente les relations entre les différents fichiers de configuration de BIND.

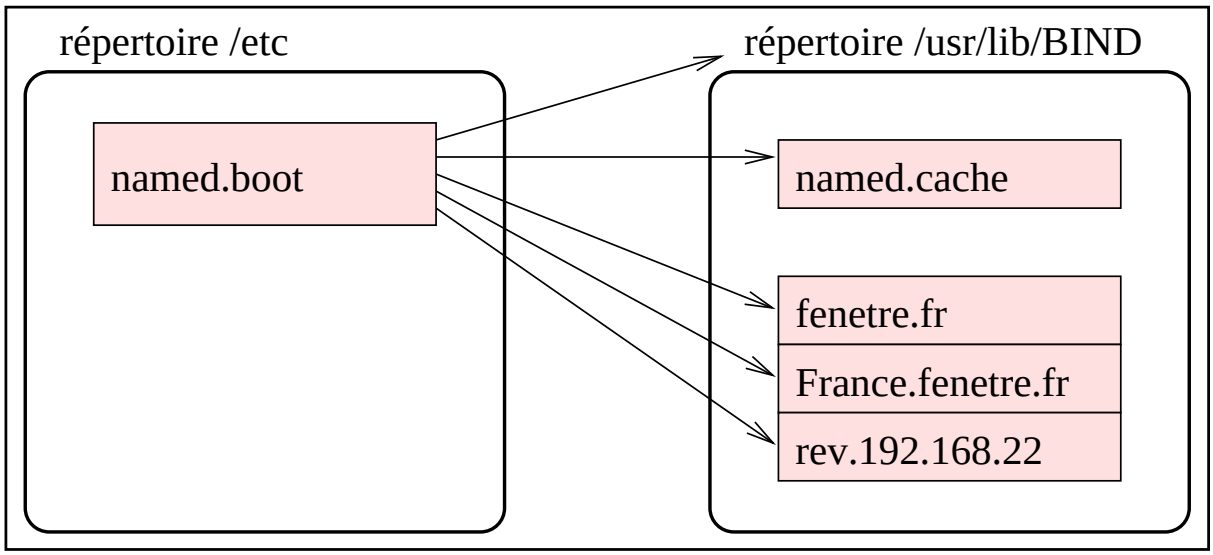

Figure 4.11 *Fichiers de configuration de BIND*

Fichier de démarrage

Le fichier de démarrage contient un ensemble de directives, une par ligne.

Les différentes primitives sont récapitulées dans le tableau 4.2 page ci-contre.

Le fichier de démarrage de `ns.fenetre.fr` présenté sur la figure 4.12 page 170 suppose que `ns.fenetre.fr` est primaire de la zone `fenetre.fr`, secondaire de la zone `Italy.fenetre.fr` et primaire de la zone inverse `22.192.168.in-addr.arpa`.

4.9.3 Fichier de cache

Le fichier de cache contient généralement la liste des adresses IP des serveurs de noms faisant autorité pour la racine, c'est-à-dire « Point ». Cette liste est fondamentale pour la première

Type	Fonction
directory	Indication du répertoire qui contient les autres fichiers, par exemple : `directory /usr/lib/BIND`
cache	Nom d'un fichier de cache, par exemple : `cache . named.cache`
primary	Être primaire d'une zone directe ou inverse. La syntaxe est la suivante : primary [domaine] [fichier de zone] par exemple : `primary fenetre.fr fenetre.fr.zone`

secondary	Être secondaire d'une zone directe ou inverse. La syntaxe est la suivante :
	`secondary [domaine] [IP maître(s)] [fichier de backup]`
	où `[IP maître(s)]` est une liste d'adresses IP de maîtres, qui souvent se réduit à l'adresse IP du primaire, par exemple :
	`secondary Italy.fenetre.fr 192.168.210.15 Italy.fenetre.fr.zone`

forwarders	Adresse des *forwarders*, par exemple : `forwarders 192.168.22.93 192.168.22.94`
xfrnets	Limitation des transferts de zones depuis une liste de machines, par exemple : `xfernets 192.168.210.15`
include	Inclusion d'un fichier de configuration annexe, par exemple : `include zones.defs`
bogusns	Liste de serveurs à ne pas contacter, car leurs données sont corrompues, par exemple : `bogusns 10.0.0.1`

check-names	Vérifier que le jeu de caractères utilisé pour les noms contenus dans les zones spécifiées est correct, et éventuellement refuser de les servir (paramètre `fail`), activer un *warning* (paramètre `warning`) ou ignorer (paramètre `ignore`); par exemple :
	`check-names primary fail`
	`check-names secondary warn`
	`check-names response ignore`

limit	Définition de limites diverses, par exemple sur la mémoire occupée par le démon ou le nombre de transferts de zones simultanés.
options	Choix d'options diverses, dont voici une liste des plus utiles : `no-recursion` : seules les requêtes non récursives sont traitées par `BIND`. Ne pas utiliser cette option avec un serveur listé dans `/etc/resolv.conf` d'un client. `query-log` : enregistrement auprès du `syslog` de toutes les requêtes qui parviennent à `BIND`. `forward-only` : passe `BIND` en mode esclave (utilisé par exemple sur le serveur DNS du réseau privé dans le cadre d'une solution firewall). Exemple : `options forward-only query-log`

Tableau 4.2 *Primitives du fichier de démarrage*

```
directory    /usr/lib/BIND

cache        .                                  named.cache

primary      fenetre.fr                         fenetre.fr.zone
primary      22.192.168.in-addr.arpa            168.192.22.rev

secondary    Italy.fenetre.fr 192.168.210.15 Italy.fenetre.fr.zone
```

Figure 4.12 *Exemple de fichier de démarrage*

phase d'une recherche récursive : `BIND` doit interroger un serveur de la racine, ce dernier le redirige alors vers les serveurs faisant autorité sur un TLD.

On peut récupérer le fichier de cache à jour à l'URL suivant :
`ftp://ftp.rs.internic.net/domain/named.root`

La figure 4.13 page ci-contre présente son contenu.

On remarque ainsi qu'il y a neuf serveurs répartis sur les cinq continents pour gérer la racine. Ces neufs serveurs sont les garants du bon fonctionnement du service de noms sur l'Internet.

4.9.4 Configuration d'une zone directe

Un fichier de zone directe contient un SOA et une suite d'enregistrements de types variés.

Par exemple, la zone `fenetre.fr` contient les données de la figure 4.14 page 172.

4.9.5 Configuration d'une zone inverse

Un fichier de zone inverse contient un SOA et une série d'enregistrements de types variés.

La figure 4.15 page 173 présente un exemple de zone inverse :
`22.168.192.in-addr.arpa`

4.9.6 Zones recommandées

Les serveurs DNS sur l'Internet sont sujets à pollution. Il suffit par exemple qu'une information incorrecte soit injectée dans un serveur du réseau pour qu'elle puisse se propager aux autres serveurs. En effet, à chaque réponse à une requête, un serveur peut fournir des informations complémentaires qui n'ont pas été demandées mais dont il sait qu'elles peuvent être utiles. Il peut s'agir par exemple de glue dont nous avons déjà vu le rôle. D'autre part, il faut éviter les requêtes inutiles, ou au moins ne pas les laisser sortir du réseau local. Les requêtes le plus souvent incorrectes concernent les enregistrements de l'adresse IP `0.0.0.0` et de

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.RS.INTERNIC.NET
;       -OR- under Gopher at    RS.INTERNIC.NET
;           under menu          InterNIC Registration Services (NSI)
;              submenu          InterNIC Registration Archives
;           file                named.root
;
;       last update:    Nov 8, 1995
;       related version of root zone:   1995110800
;
; formerly NS.INTERNIC.NET
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     128.9.0.107
;
; formerly C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                        3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.      3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                        3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                        3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                        3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A     192.36.148.17
; End of File
```

Figure 4.13 *Fichier de cache*

```
@               IN SOA          ns.fenetre.fr. ls.fenetre.fr. (
                                1996120101      ;Serial
                                28800           ;Refresh (8 heures)
                                7200            ;Retry   (2 heures)
                                604800          ;Expire  (7 jours)
                                86400 )         ;Minimum (1 jour)

; Liste des serveurs faisant autorité
                IN NS           ns.fenetre.fr.
                IN NS           ns.fournisseur.fr.

; Déclaration des réseaux appartenant à l'organisme fenetre.fr
                IN PTR          0.22.168.192.in-addr.arpa.
                IN PTR          0.0.16.172.in-addr.arpa.

; Acheminement du courrier pour les messages du type user@fenetre.fr
                IN MX      10   mail.fenetre.fr.
                IN MX     100   mail.fournisseur.fr.

; Acheminement du courrier pour les messages du type user@xyz.fenetre.fr
*               IN MX      10   mail.fenetre.fr.
                IN MX     100   mail.fournisseur.fr.

; Déclaration des noms des réseaux ls-net.fenetre.fr et ls-net-2.fenetre.fr
ls-net          IN PTR          0.22.168.192.in-addr.arpa.
ls-net-2        IN PTR          0.0.16.172.in-addr.arpa.

; Informations sur la passerelle
gw              IN A            192.168.22.35
                IN TXT          "Routeur du site"
                IN HINFO        SUN-IPC Solaris-2.4
gw-ppp          IN A            192.168.200.2
                IN TXT          "Routeur du site"
                IN HINFO        SUN-IPC Solaris-2.4

; Acheminement du courrier pour les messages du type gw@fenetre.fr
gw              IN MX      10   mail.fenetre.fr.
                IN MX     100   mail.fournisseur.fr.

; Acheminement du courrier pour les messages du type gw-ppp@fenetre.fr
gw-ppp          IN MX      10   mail.fenetre.fr.
                IN MX     100   mail.fournisseur.fr.

; Adresse de la passerelle SMTP
mail            IN A            192.168.22.36
                IN TXT          "Machine appartenant a Stoned"
                IN HINFO        SUN-IPC Solaris-2.5

; Acheminement du courrier pour les messages du type mail@fenetre.fr
                IN MX      10   mail.fenetre.fr.
                IN MX     100   mail.fournisseur.fr.

; Alias pour le serveur WWW sur gw.fenetre.fr
www             IN CNAME        gw.fenetre.fr.

; Délégation du sous-domaine Italy.fenetre.fr
Italy           IN NS           ns.Italy.fenetre.fr.
                IN NS           ns.fournisseur-italien.it.

; Ajout de la glue pour la délégation d'Italy.fenetre.fr
ns.Italy.fenetre.fr.   IN A     192.168.210.15
```

Figure 4.14 *Exemple de zone directe*

```
@               IN SOA          ns.fenetre.fr. ls.fenetre.fr. (
                                1996120201      ;Serial
                                28800           ;Refresh (8 heures)
                                7200            ;Retry   (2 heures)
                                604800          ;Expire  (7 jours)
                                86400 )         ;Minimum (1 jour)

; Liste des serveurs faisant autorité
                IN NS           ns.fenetre.fr.
                IN NS           ns.fournisseur.fr.

; nom associé à 192.168.22.35
35              IN PTR          gw.fenetre.fr.

; nom associé à 192.168.22.36
36              IN PTR          mail.fenetre.fr.

; nom du réseau 192.168.22.0 et masque de sous-réseaux
0               IN PTR          ls-net.fenetre.fr.
                IN A            255.255.255.224
```

Figure 4.15 *Exemple de zone inverse*

l'adresse de diffusion `255.255.255.255`. Pour éviter leur propagation, le RFC 1912 demande aux administrateurs de gérer sur tout serveur de noms les deux zones suivantes :

- `0.in-addr.arpa`
- `255.in-addr.arpa`

De plus, pour éviter de charger le réseau avec les requêtes liées à l'interface *loopback*, ce RFC propose aussi que tous les serveurs DNS fassent autorité pour les zones `localhost` et `0.0.127.in-addr.arpa`.

Il faut donc ajouter les données suivantes au fichier `/etc/named.boot` :

```
primary         localhost                       localhost
primary         0.0.127.in-addr.arpa            127.0
primary         255.in-addr.arpa                255
primary         0.in-addr.arpa                  0
```

Le fichier `/usr/local/BIND/localhost` de `ns.fenetre.fr` contient donc :

```
@               IN SOA          ns.fenetre.fr. ls.fenetre.fr. (
                                1996120201      ;Serial
                                28800           ;Refresh (8 heures)
                                7200            ;Retry   (2 heures)
                                604800          ;Expire  (7 jours)
                                86400 )         ;Minimum (1 jour)

localhost.      IN A            127.0.0.1
```

Le fichier `/usr/lib/BIND/127.0` contient :

```
@               IN SOA          ns.fenetre.fr. ls.fenetre.fr. (
                                1996120201      ;Serial
                                28800           ;Refresh (8 heures)
                                7200            ;Retry   (2 heures)
                                604800          ;Expire  (7 jours)
                                86400 )         ;Minimum (1 jour)

1               IN PTR  localhost.
```

Le fichier `/usr/lib/BIND/255` contient :

```
@               IN SOA          ns.fenetre.fr. ls.fenetre.fr. (
                                1996120201      ;Serial
                                28800           ;Refresh (8 heures)
                                7200            ;Retry   (2 heures)
                                604800          ;Expire  (7 jours)
                                86400 )         ;Minimum (1 jour)

                IN NS           ns.fenetre.fr.
```

Le fichier `/usr/lib/BIND/0` contient :

```
@               IN SOA          ns.fenetre.fr. ls.fenetre.fr. (
                                1996120201      ;Serial
                                28800           ;Refresh (8 heures)
                                7200            ;Retry   (2 heures)
                                604800          ;Expire  (7 jours)
                                86400 )         ;Minimum (1 jour)

                IN NS           ns.fenetre.fr.
```

4.9.7 Activation du démon BIND

Pour activer le démon BIND, il suffit d'appeler `/usr/sbin/in.named` depuis un fichier de démarrage de la machine.

S'il est appelé sans option, le démon BIND lit le fichier `/etc/named.boot` et en déduit la localisation des fichiers de cache et de zones.

On peut changer la localisation du fichier de démarrage par défaut et d'autres paramètres avec les options de la ligne de commande que voici :

Option	Fonction
`-d debuglevel`	Information de *debugging*
`-p port1[/port2]`	Imposer d'autres ports que le port standard ; `port1` est utilisé pour contacter d'autres serveurs et le port d'écoute est `port2` (option utilisée en phase de *debugging*)
`-b bootfile`	Utiliser un autre fichier que `/etc/named.boot`

4.10 Le NIC-France

Depuis 1987, l'INRIA administre le domaine `fr`. En 1992, le NIC-France (Network Information Center) est créé pour assurer une triple mission :

- mise en place d'une charte d'attribution de noms ;
- gestion de la zone `fr` : mise à disposition de serveurs de noms pour cette zone et délégations de sous-domaines ;
- coordination avec les autres NIC et maintenance des serveurs de noms pour les zones inverses qui lui ont été déléguées.

Sur le serveur WWW du NIC-France, `http://www.nic.fr`, on peut trouver de nombreux renseignements sur l'Internet, les procédures et services du NIC-France, des statistiques diverses, la liste des autres NIC, etc.

4.10.1 Charte d'attribution de noms

La charte d'attribution de noms définie par le NIC-France précise les règles d'attribution de sous-domaines de `fr`. Lorsqu'une société a obtenu un sous-domaine de `fr`, elle peut y créer des sous-domaines selon ses propres règles, la charte d'attribution de noms ne s'y appliquant pas, excepté pour la règle suivante :

> Un domaine qui appartient à une entité administrative et juridique ne peut être utilisé par une autre entité administrative et juridique (société-B.société-A.fr est interdit).

Le nom de domaine demandé peut être le nom de l'organisme ou de la société, son sigle ou une marque qui lui appartient.

Documents nécessaires pour l'attribution d'un domaine sous `fr`

Pour obtenir l'attribution d'un nom de domaine, il faut faire parvenir à son fournisseur Internet un exemplaire rempli et signé du formulaire de création de domaine disponible à l'URL :
`ftp://ftp.nic.fr/pub/formulaires/NIC/Formulaire-Domaine-Fr.ps`.
Le fournisseur se charge alors de contacter le NIC-France pour demander la délégation du domaine qui est alors mis en place sous 48 heures.
S'il s'agit d'un nom de société, un extrait du Kbis et le numéro de SIRET est demandé.
S'il s'agit d'une association, une copie de la parution au Journal Officiel ou le récépissé de déclaration à la préfecture est demandé. La délégation est alors mise en place sous le domaine `asso.fr`.
S'il s'agit d'une marque, il faut fournir le certificat d'enregistrement à l'INPI ainsi que son numéro. La délégation est alors mise en place sous le domaine `tm.fr` (*trademark*).
S'il s'agit d'une publication, il faut fournir une copie du document émanant de la Bibliothèque Nationale et portant le numéro ISSN.

C'est le fournisseur qui fait la demande de délégation pour son client. Le NIC-France lui

impute alors un coût en fonction de la convention qu'ils ont signée ensemble. Le fournisseur a le choix entre deux options :

- option 1 : le fournisseur adhère au service NIC et peut ainsi participer au comité de concertation. Il doit pour cela verser 30 kF et toute création de domaine lui est alors facturée 800 F (coût 1996), montant pondéré par un coefficient 0,7, 1 ou 3 en fonction du support technique fourni par le NIC ;
- option 2 : le fournisseur n'adhère pas au service NIC ; toute création de domaine lui est facturée 2 400 F (coût 1996).

Le nom de domaine doit posséder au moins trois caractères, sauf conditions exceptionnelles, comme par exemple `m6.fr`. De plus, il existe un certain nombre de noms réservés qui ne peuvent pas être attribués, par exemple `atm.fr`, `ftp.fr`, `internet.fr`, etc.

Le tableau 4.3 page 187 définit la charte d'attribution de noms.

4.10.2 Service de test des zones

Pour mettre en place une zone, il faut mettre en place au moins deux serveurs pour la gérer (un primaire et un secondaire), de préférence sur des accès Internet distincts. Les organismes connectés par une liaison permanente se chargent le plus souvent du primaire et laissent à leur fournisseur le soin de gérer un secondaire. Les organismes connectés de façon intermittente ne peuvent gérer ni le primaire ni le secondaire, leur fournisseur se charge alors de mettre en place ces deux serveurs sur son réseau privé.

Une fois ces serveurs mis en place, le NIC-France modifie le contenu de la zone `fr` afin de déléguer le domaine attribué. Pour cela, il ajoute des enregistrements de type NS dans le fichier de zone de `fr`.

Le NIC-France impose un certain nombre de conditions techniques avant de mettre en place la délégation, par exemple sur les paramètres du SOA ou d'autres valeurs contenues dans le fichier de la zone à déléguer.

Ces conditions peuvent être vérifiées rapidement à l'aide d'un automate très pratique, nommé ZoneCheck et accessible sur le serveur WWW du NIC-France. Cet automate se connecte sur les serveurs de noms de la zone, effectue les vérifications et produit un rapport signalant les éventuels problèmes détectés.

Pour l'utiliser, il suffit de se connecter avec un navigateur World Wide Web sur l'URL `http://www.nic.fr/ZoneCheck/`. Après avoir choisi la langue (anglais ou français), l'écran de la figure 4.16 page ci-contre apparaît. On saisit le nom de domaine à tester et on choisit « Chercher... ». On entre alors, dans le formulaire de la figure 4.17 page 178, les adresses IP des serveurs de noms faisant autorité pour la zone et qu'on veut tester, et on choisit « Vérifier... ». L'automate affiche enfin, en moins de trente secondes, les différents problèmes éventuels concernant l'installation de cette zone.

On suit les indications pour faire des corrections et on répète alors l'opération jusqu'à avoir résolu l'ensemble des problèmes.

Figure 4.16 *Service ZoneCheck*

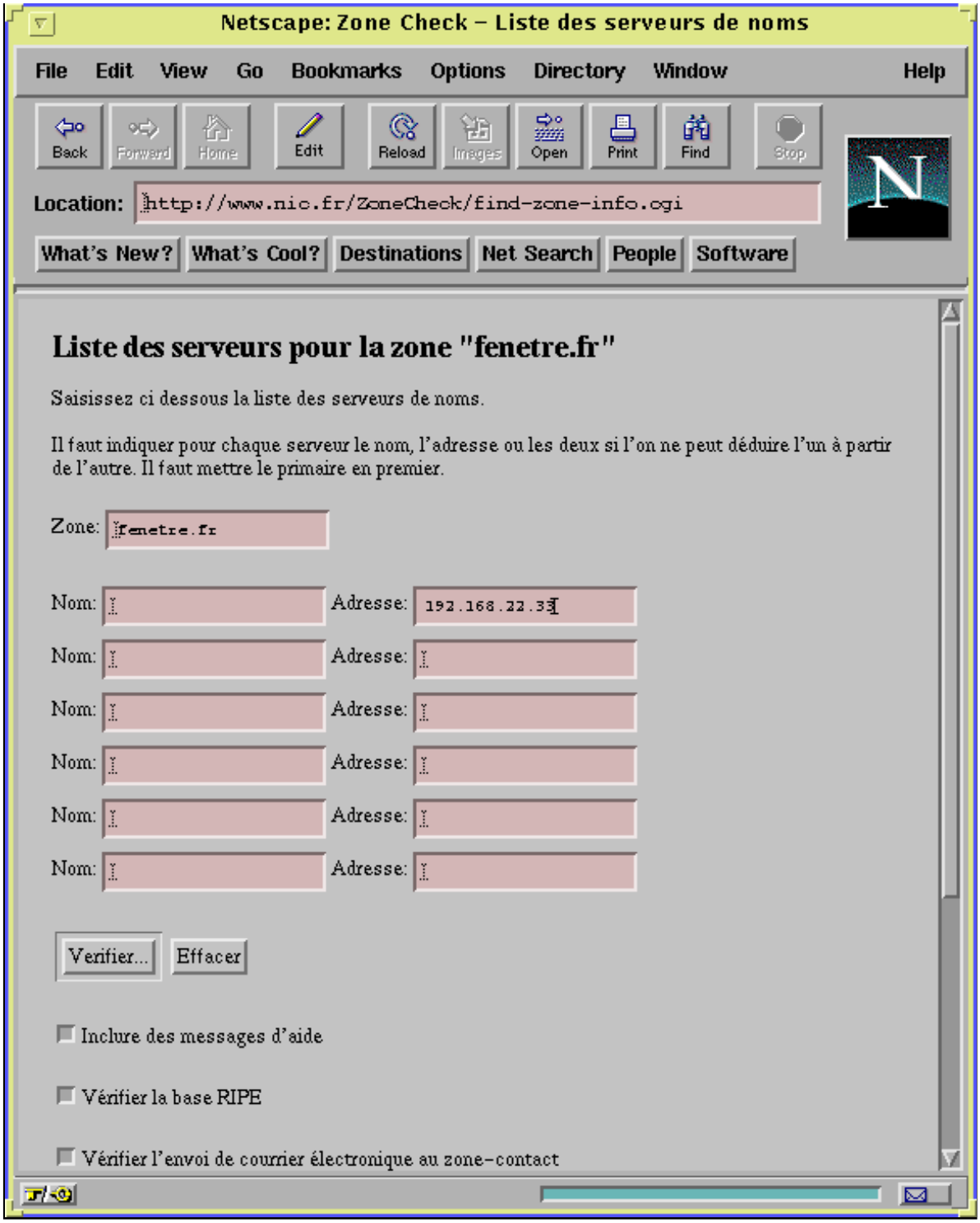

Figure 4.17 *Service ZoneCheck*

4.11 Attribution d'un domaine sous `com`, `org`, `edu` et `net`

Un des nombreux services offerts par l'InterNIC consiste à déléguer des sous-domaines des TLD suivants : `com`, `org`, `edu` ou `net`.

Pour obtenir une telle délégation, il faut tout d'abord mettre en place un primaire et un ou plusieurs secondaires et récupérer le formulaire de demande de domaine de l'InterNIC par FTP à l'URL suivant :

```
ftp://ftp.rs.internic.net/templates/domain-template.txt
```

Une fois rempli, il faut l'envoyer à `hostmaster@internic.net`.

Une autre méthode consiste à utiliser un navigateur World Wide Web, et à se connecter à l'URL `http://rs.internic.net/reg/domain-forms.html` qui contient un formulaire HTML présentant les mêmes champs que le formulaire de demande de domaine, de remplir ce formulaire et de le valider.

Le serveur WWW envoie alors à l'utilisateur, par courrier électronique, le formulaire de demande de domaine rempli conformément aux indications qui ont été fournies. Il suffit alors de vérifier ces informations et de renvoyer ce message à `hostmaster@internic.net`.

Le coût de l'opération est de 100 $. Il couvre deux ans de délégation du domaine. Après cette période, l'InterNIC renvoie tous les ans une facture de 50$.

Avec un cours du dollar à cinq francs, on constate que trois ans de délégation sous `com` coûtent sensiblement aussi cher qu'une délégation définitive par le NIC-France sous `fr` (800 F pour un domaine si le fournisseur adhère au service NIC).

Il est donc souvent plus économique à long terme de demander une délégation sous `fr` que sous `com`.

4.12 Attribution d'un domaine sous `eu.org`

L'administrateur du domaine `eu.org`, Pierre BEYSSAC, fournit un service gratuit de délégation sous ce domaine. Il s'occupe ainsi des serveurs de noms qui gèrent ce domaine, et prend en charge les frais de délégation auprès de l'InterNIC.

La charte d'enregistrement est disponible sur `http://ns.eu.org/fr/policy.html` et le formulaire d'inscription est accessible sur `http://ns.eu.org/fr/form.html`.

Ce service gratuit s'adresse en priorité aux associations et aux particuliers, mais les sites commerciaux sont aussi accueillis.

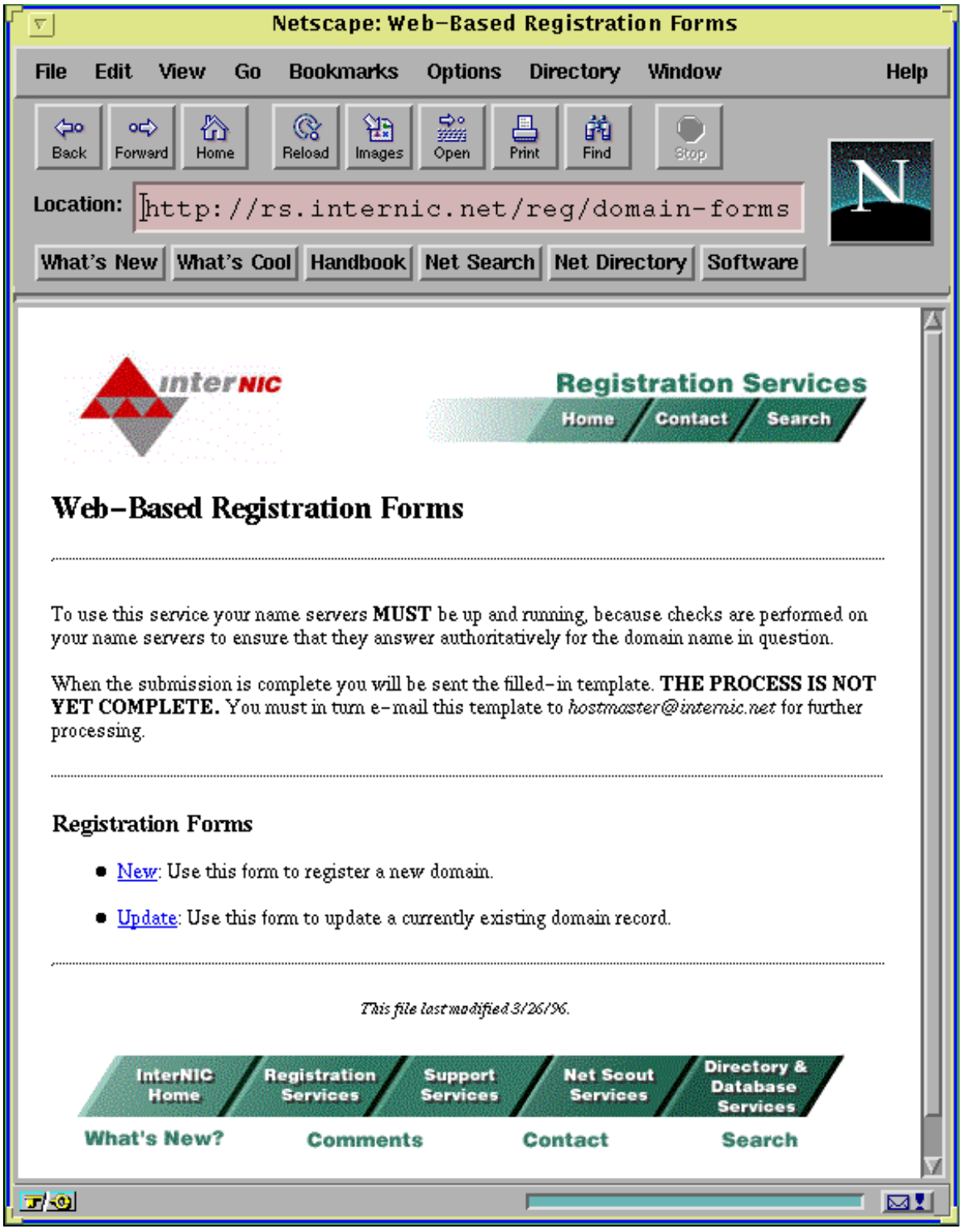

Figure 4.18 *Délégation d'un domaine auprès de l'InterNIC*

4.13 Les bases *Whois*

Des informations administratives diverses sont maintenues sur des bases *Whois*. On y trouve notamment des informations sur les domaines, adresses de réseaux, annonces de routes, noms de contacts techniques ou administratifs saisis dans les différents formulaires de demande de délégation, etc. Cela permet par exemple de vérifier que le sous-domaine de `com` ou `fr` qu'on désire réserver est bien libre.

Pour interroger ces bases, on utilise la commande `whois`.

Voici quelques-uns des types d'objets présents dans ces bases :

- **objet de type domaine (domain)** : informations techniques sur un domaine ;
- **objet de type réseau (inetnum)** : informations techniques sur la délégation d'une adresse réseau ;
- **objet de type route (route)** : informations sur un ensemble de réseaux annoncés sur l'Internet au sein d'une route de type CIDR par un protocole de passerelle externe ;
- **objet de type personne (person)** : informations sur un contact technique ou administratif référencé dans un autre type d'objet ;
- **objet de type système autonome (aut-num)** : informations techniques sur un fournisseur.

Les principaux serveurs *Whois* sont les suivants :

- `whois.internic.net` : ce serveur regroupe des objets de nombreux types excepté le type route. Il rassemble au sein d'un même serveur les informations contenues dans d'autres serveurs *Whois* ;
- `whois.ra.net` : ce serveur regroupe tous les objets de type route de l'Internet. C'est à partir de sa base que les tables de routage des NAP sont établies ;
- `whois.ripe.net` : ce serveur regroupe des objets de tous types pour l'Europe ;
- `whois.apnic.net` : ce serveur regroupe les objets de type réseau pour la zone Asie-Pacifique ;
- `whois.nic.fr` : ce serveur maintenu par le NIC-France contient des objets concernant la France, notamment des informations sur les sous-domaines de `fr`.

Quand on veut obtenir une information particulière, il faut déterminer le type d'objet et la zone géographique associés, afin de savoir quel serveur *Whois* interroger.

Ce sont les fournisseurs et les NIC qui mettent à jour les informations de ces bases.

Effectuons quelques interrogations sur les bases *Whois*. Pour cela on utilise l'utilitaire `whois` disponible pour Unix à l'URL suivant :

```
ftp://ftp.nic.fr/pub/programmes/RIPE/ripe-whois-tools-2.0.tar.gz
```

Notons qu'il en existe aussi sur PC et Macintosh.

Sous Unix, l'option `-h` permet de préciser le nom du serveur *Whois*.

Nous désirons tout d'abord vérifier que le domaine *fenetre.fr* n'est pas réservé, afin de demander son attribution auprès du NIC-France :

```
<ls@fenetre> whois -h whois.nic.fr fenetre.fr
No entries found for the selected source(s).
[...]
```

Si le domaine avait été réservé, on aurait eu le type d'informations présenté sur la figure 4.19.

```
<ls@fenetre> whois -h whois.nic.fr enst.fr
domain:      enst.fr
descr:       Ecole Nationale Superieure des Telecommunications - Telecom/Paris
descr:       46, rue Barrault, F-75634 Paris CEDEX 13
admin-c:     Jean-Pierre Bach
tech-c:      Philippe Dax
tech-c:      Guillaume Gerard
zone-c:      AR41
nserver:     inf.enst.fr
nserver:     tyner.res.enst.fr
nserver:     dameron.res.enst.fr
nserver:     layon.inria.fr
dom-net:     137.194.0.0
dom-net:     192.33.155.0
mnt-by:      FR-NIC-MNT
changed:     Benoit.Grange@inria.fr 950307
source:      RIPE

person:      Jean-Pierre Bach
address:     Ecole Nationale Superieure Telecommunications
address:     Centre de Calcul
address:     46, rue Barrault, F-75634 Paris CEDEX 13, France
phone:       +33 1 45 81 42 96
e-mail:      bach@enst.enst.fr
changed:     Annie.Renard@inria.fr 940209
source:      RIPE

[...]
```

Figure 4.19 *Exemple d'utilisation de `whois` pour un domaine*

On constate que le serveur *Whois* nous fournit, en plus de l'objet domaine, les objets personne associés.

Nous pouvons maintenant essayer d'obtenir des informations sur le réseau `137.194.0.0`. La figure 4.20 page suivante présente le type de résultat obtenu.

En plus des contacts techniques, deux objets ont été retournés : il s'agit de l'objet de type inetnum `137.194.0.0` qui correspond à la délégation de ce réseau, et de l'objet de type route `137.194.0.0/16` qui correspond à l'annonce de l'accessibilité de ce réseau. À cet objet est associé l'objet AS1717 de type système autonome (aut-num). Il s'agit du fournisseur Renater qui connecte ce réseau à l'Internet. Pour obtenir des informations sur ce fournisseur portant le numéro de système autonome 1717, il suffit d'interroger par exemple

```
<ls@fenetre> whois -h whois.nic.fr 137.194.0.0
inetnum:     137.194.0.0
netname:     ENST-NET
descr:       Ecole Nationale Superieure Telecommunications
descr:       Centre de Calcul
descr:       46, rue Barrault, 75634 Paris CEDEX 13, France
country:     FR
admin-c:     Jean-Pierre Bach
tech-c:      Philippe Dax
remarks:     FNET
changed:     Annie.Renard@inria.fr 940209
source:      RIPE

route:       137.194.0.0/16
descr:       RENATER
descr:       Universite Pierre et Marie Curie
descr:       4 place Jussieu 75252 PARIS CEDEX 05
descr:       FRANCE
origin:      AS1717
mnt-by:      AS1717-MNT
changed:     rensvp@renater.fr 951212
source:      RIPE

person:      Jean-Pierre Bach
address:     Ecole Nationale Superieure Telecommunications
address:     Centre de Calcul
address:     46, rue Barrault, F-75634 Paris CEDEX 13, France
phone:       +33 1 45 81 42 96
e-mail:      bach@enst.enst.fr
changed:     Annie.Renard@inria.fr 940209
source:      RIPE

[...]
```

Figure 4.20 *Exemple d'utilisation de `whois` pour un réseau*

`whois.ra.net` comme indiqué sur la figure 4.21 page suivante.

Les champs `as-in` et `as-out` indiquent à quels autres fournisseurs Renater annonce ses routes, et de quels autres fournisseurs il prend des annonces. Il s'agit donc de la politique de routage de ce fournisseur. Pour connaître les noms de fournisseurs associés aux numéros de systèmes autonomes indiqués ici, il suffit là encore d'utiliser *Whois*.

Enfin, pour connaître des informations sur les TLD, il faut effectuer une interrogation sur l'objet *TLD-dom*. Par exemple, pour le domaine `com`, on interroge `whois.internic.net` comme indiqué sur la figure 4.22 page suivante.

4.14 Outils de tests

Les deux principaux outils d'interrogation du DNS fournis avec `BIND` sont `nslookup` et `dig`.

Ils permettent tous les deux d'émettre tous types de requêtes vers un serveur local ou distant, c'est la syntaxe d'utilisation qui les distingue profondément. Après avoir lancé le programme

```
<ls@fenetre> whois -h whois.ra.net AS1717
aut-num:     AS1717
descr:       RENATER
descr:       Reseau National de telecommunications pour la Technologie
descr:       l'Enseignement et la Recherche
descr:       FR
as-in:       from AS1755 100 accept ANY
as-in:       from AS5511 100 accept ANY
as-in:       from AS2470 100 accept AS2470
as-in:       from AS789 50 accept AS789
as-in:       from AS786 50 accept AS786
as-in:       from AS1899 50 accept AS1899
as-in:       from AS2917 50 accept AS2917
as-in:       from AS3215 50 accept AS3215
as-out:      to AS5511 announce AS1717
as-out:      to AS2470 announce AS1717
as-out:      to AS1755 announce AS1717 AS2470
as-out:      to AS786 announce AS1717
as-out:      to AS789 announce AS1717
as-out:      to AS1899 announce AS1717
as-out:      to AS2917 announce AS1717
as-out:      to AS3215 announce AS1717
default:     AS1755 10
admin-c:     Michel Lartail
tech-c:      Isabelle Morel
tech-c:      Marie-Helene Guilmin
mnt-by:      AS1717-MNT
changed:     rensvp@renater.fr 960423
source:      RIPE
```

Figure 4.21 *Exemple d'utilisation de `whois` pour un système autonome*

```
<ls@fenetre> whois -h whois.internic.net com-dom
Commercial top-level domain (COM-DOM)
   Network Solutions, Inc.
   505 Huntmar park Dr.
   Herndon, VA 22070

   Domain Name: COM

   Administrative Contact, Technical Contact, Zone Contact:
      Network Solutions, Inc.  (HOSTMASTER)  hostmaster@INTERNIC.NET
      (703) 742-4777 (FAX) (703) 742-4811

   Record last updated on 02-Sep-94.
   Record created on 01-Jan-85.

   Domain servers in listed order:

   A.ROOT-SERVERS.NET           198.41.0.4
   H.ROOT-SERVERS.NET           128.63.2.53
   B.ROOT-SERVERS.NET           128.9.0.107
   C.ROOT-SERVERS.NET           192.33.4.12
   D.ROOT-SERVERS.NET           128.8.10.90
   E.ROOT-SERVERS.NET           192.203.230.10
   I.ROOT-SERVERS.NET           192.36.148.17
   F.ROOT-SERVERS.NET           192.5.5.241
   G.ROOT-SERVERS.NET           192.112.36.4

[...]
```

Figure 4.22 *Exemple d'utilisation de `whois` pour un TLD*

`nslookup`, on peut entrer successivement différentes commandes pour mettre en forme la requête, sélectionner le serveur cible, définir un niveau de *debug*, etc. À l'inverse, `dig` prend toutes ces informations en paramètres de la ligne de commande.

Examinons par exemple comment on peut interroger `ns.fenetre.fr` pour connaître le nom associé à l'adresse IP `192.168.22.33`.

Avec `dig`, il suffit d'entrer la commande suivante :

```
<ls@gw> dig @ns.fenetre.fr 33.22.168.192.in-addr.arpa. PTR

; <<>> DiG 2.0 <<>> @inf.enst.fr 33.22.168.192.in-addr.arpa. PTR
;; ->>HEADER<<- opcode: QUERY , status: NOERROR, id: 10
;; flags: qr rd ra ; Ques: 1, Ans: 2, Auth: 2, Addit: 2
;; QUESTIONS:
;;      33.22.168.192.in-addr.arpa, type = PTR, class = IN

;; ANSWERS:
33.22.168.192.in-addr.arpa.       136071  PTR     gw.fenetre.fr.

;; Sent 1 pkts, answer found in time: 6 msec
;; FROM: gw to SERVER: ns.fenetre.fr 192.168.22.34
;; WHEN: Fri Sep 13 20:55:00 1996
;; MSG SIZE  sent: 44  rcvd: 180

<ls@gw>
```

Avec `nslookup`, on procède comme ceci :

```
<ls@gw> nslookup
Default Server:  ns2.fenetre.fr
Address:  192.168.22.37

> lserver ns.fenetre.fr
Default Server:  ns.fenetre.fr
Address:  192.168.22.34

> set type=PTR
> 33.22.168.192.in-addr.arpa.
Server:  ns.fenetre.fr
Address:  192.168.22.34

33.22.168.192.in-addr.arpa        name = gw.fenetre.fr
> exit
<ls@gw>
```

4.15 Automates d'analyse DNS

De très nombreux outils d'analyse automatique de la configuration des DNS sont disponibles gratuitement sur le réseau. On peut par exemple en trouver un bon nombre à l'URL `ftp://ftp.ibp.fr/pub/network/dns/`.

Citons parmi eux deux des plus utilisés : il s'agit de `dnswalk` et `lamers`.

4.15.1 `dnswalk`

`dnswalk` effectue des transferts des zones des domaines spécifiés en argument et procède à de nombreux tests de cohérence sur leur contenu. C'est donc un outil de vérification des zones qui utilise directement le DNS pour aller rechercher l'information à vérifier. Il faut installer `perl` et `dig` sur la machine où l'on veut l'utiliser. Il vérifie par exemple que si `x.fenetre.fr` possède un enregistrement de type A de valeur `192.168.22.1`, il existe alors aussi un enregistrement de type PTR pour `1.22.168.192.in-addr.arpa` de valeur `x.fenetre.fr`. On peut lui fournir différents paramètres sur la ligne de commande, afin par exemple de lui indiquer les différents tests à mettre en jeu.

En voici un exemple d'utilisation :

```
<ls@fenetre> ./dnswalk enst.fr.
Getting zone transfer of enst.fr. from inf.enst.fr....done.
Checking enst.fr.
SOA=inf.enst.fr.        contact=hostmaster.enst.fr.
 demo52.enst.fr. A 137.194.6.52: no PTR record
 res.enst.fr. NS gmuvax2.gmu.edu.: CNAME (to portal.gmu.edu)
 bretagne.enst.fr. CNAME enst-bretagne.fr.: unknown host
 broadcast-inf.enst.fr. A 137.194.161.255: no PTR record
 enst-netmask.enst.fr. A 255.255.254.0: no PTR record
 www-ima.enst.fr. CNAME mathieu.enst.fr.: CNAME (to ima.enst.fr)
 gw-x25.enst.fr. A 193.49.49.233: no PTR record
 maisel2-guyton.enst.fr. A 137.194.8.254: no PTR record
 khamsin-fddi.enst.fr. A 137.194.202.130: no PTR record
```

4.15.2 `lamers`

L'utilitaire `lamers` analyse les fichiers de *log* produits par `BIND` pour détecter des **Lame Delegations**. Il s'agit de délégations incorrectes de zones où des machines se retrouvent listées en tant que serveurs faisant autorité pour une zone alors qu'elles ne semblent pas en assurer la gestion. L'utilitaire `lamers` envoie alors un courrier électronique comme suit :

```
This message was machine generated.  It is intended to alert you to a
possible problem with one of the nameservers for the domain listed
below.  If you have any questions about this message, please contact
dns-maintenance@umich.edu.

The nameserver listed in the subject (and again below) was detected as
a "lame delegation" by the UMICH.EDU nameservers.

        Domain: fenetre.fr
        Server: 192.168.22.34

The following paragraphs describe what a lame delegation is and
suggests some things you might do to eliminate the lame delegation.  If
you are an experienced hostmaster, you probably do not to read the rest
of this message; if you don't know what a lame delegation is, you
probably want to keep reading.
```

Entité	**format**	**Exemple**
Académies	ac-*nom*.fr	ac-lyon.fr
Ambassades	amb-*nom*.fr	amb-wash.fr
Assistance publique	ap-*nom*.fr	ap-hop-paris.fr
Associations	*nom*.asso.fr	xyz.asso.fr
Avocats	*nom*.barreau.fr	xyz.barreau.fr
Ecoles d'architecture et laboratoires	*nom*.archi.fr	xyz.archi.fr
Bibliothèques municipales	bm-*ville*.fr	bm-lyon.fr
Centres hospitaliers universitaires	chu-*ville*.fr	chu-rouen.fr
Chambres de Commerce et de l'Industrie	*nom*.cci.fr	essonne.cci.fr
Commissaires-priseurs	*nom*.encheres.fr	xyz.encheres.fr
Conseils généraux	cg*département*.fr	cg13.fr (Bouches du Rhônes)
Districts	district-*nom*.fr	district-parthenay.fr (district de Parthenay)
Gouvernements et ministères	*nom*.gouv.fr	social.gouv.fr (Ministère du Travail et des Affaires Sociales)
Instituts universitaires de technologie	iut-*nom*.fr	iut-lannion.fr
Instituts Universitaires de Formation des Maîtres	*nom*.iufm.fr	xyz.iufm.fr
Mairies	mairie-*ville*.fr	mairie-metz.fr
Marques	*nom*.tm.fr	xyz.tm.fr
Technopoles	tech-*nom*.fr	tech-quimper.fr
Universités	univ-*nom*.fr u-*nom*.fr	univ-rennes1.fr (Université de Rennes1) u-grenoble3.fr (Université de Grenoble3)

Tableau 4.3 *Charte d'attribution de noms*

5

La messagerie

L'échange de courrier électronique entre différentes machines est régi par différents RFC (Request For Comments) dont la liste est donnée dans le tableau 5.1. Un logiciel capable d'échanger du courrier n'est pas obligé de se conformer à toutes ces recommandations, mais lorsqu'il prétend disposer de certaines caractéristiques, elles doivent suivre les recommandations correspondantes.

RFC	Titre
821	SMTP protocol
822	Mail header format
974	MX routing
987	Mapping between RFC822 and X.400
1049	Content-Type header field (extension du RFC 822)
1123	Host requirements (modifie les RFC 821, 822 et 974)
1344	Implications of MIME for Internet Mail Gateways
1413	Identification server
1425	SMTP Service Extensions (ESMTP)
1426	SMTP Service Extension for 8bit-MIME transport
1427	SMTP Service Extension for Message Size Declaration
1428	Transition of Internet Mail from Just-Send-8 to 8-bit SMTP/MIME
1521	MIME: Multipurpose Internet Mail Extensions

Tableau 5.1 *RFC définissant l'échange de messages*

5.1 La matrice

Tout d'abord, corrigeons une idée fausse fréquemment répandue : le courrier électronique n'est pas l'apanage du réseau Internet. Les premiers ordinateurs à s'échanger du courrier ne « parlaient » pas TCP/IP et ne se connectaient les uns aux autres que de façon intermittente, c'est-à-dire notamment que l'ensemble des ordinateurs capables de s'échanger du courrier électronique ne formaient jamais (et ne forment toujours pas) un graphe connexe à un instant donné.

On désigne habituellement sous le nom de **matrice**[1] l'ensemble des machines capables d'échanger du courrier électronique avec les ordinateurs présents sur l'Internet (et, par extension, avec tous les autres ordinateurs de la matrice). Comme le montre la figure 5.1, cet ensemble comprend tous les ordinateurs connectés à l'Internet (à savoir aussi bien l'Internet « contributif », qui regroupe les ordinateurs mettant des services à la disposition des autres que l'Internet « passif » qui ne fait qu'utiliser des services existants) ainsi que d'autres, qui ne font pas partie de l'Internet mais qui sont tout aussi capables d'envoyer et de recevoir du courrier électronique.

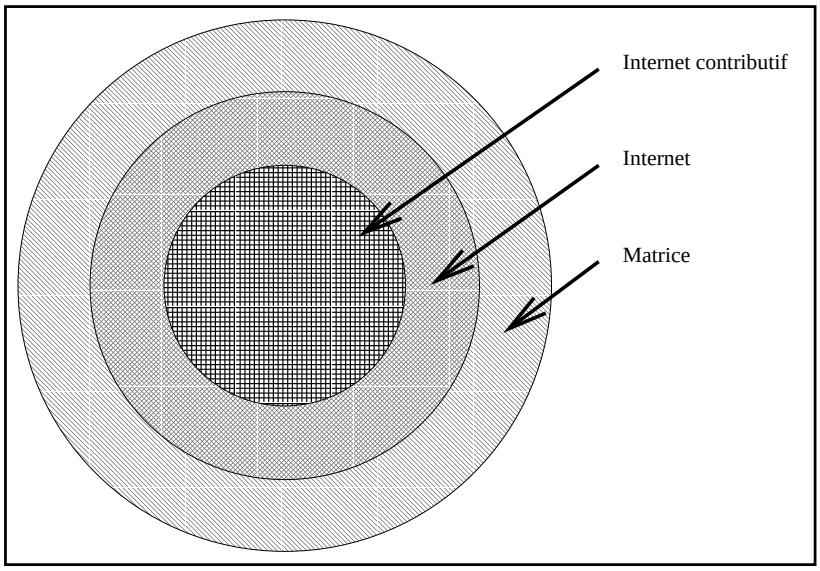

Figure 5.1 *L'Internet et la matrice*

Les ordinateurs de la matrice qui n'appartiennent pas à l'Internet comprennent les BBS[2] ainsi que les machines appartenant à d'autres réseaux comme Fidonet ou Bitnet. Ils échangent du courrier électronique en utilisant différents protocoles tels que SMTP (Simple Mail Transport Protocol), UUCP (Unix to Unix CoPy), X400 ou encore des passerelles vers des protocoles propriétaires.

1. Le terme « matrice » semble avoir été inventé par William GIBSON dans son ouvrage de science-fiction *Neuromancien* écrit en 1984.

2. BBS : *Bulletin Board Service*, ordinateurs généralement à accès libre sur lesquels on peut se connecter par le biais d'un modem et d'un ordinateur personnel.

Pour des raisons de sécurité, il arrive que certaines entreprises s'insèrent dans la matrice sans mettre en place de connexion permanente à l'Internet ; en effet, étant donné qu'un ordinateur non connecté à Internet mais appartenant à la matrice n'échange *a priori* que des messages textuels, il est difficile d'imaginer un acte de piratage envers une telle machine.

5.2 Les boîtes aux lettres

L'échange de courrier électronique entre les machines est une chose, l'échange de messages entre un utilisateur et un ordinateur en est une autre. Comme pour le courrier postal, il existe plusieurs modèles de livraison et de consultation du courrier électronique.

Les trois modes les plus courants sont le **spool**, la méthode **POP** et la méthode **IMAP**, offrant chacun ses avantages et ses inconvénients. Bien sûr, selon la méthode choisie, l'utilisateur devra se procurer un logiciel capable d'utiliser cette méthode ; de tels logiciels sont appelés « lecteurs de courrier électronique » ou MUA (Mail User Agent).

5.2.1 Le spool

Le spool (quelquefois traduit par «spoule» en français) est en fait un espace disque, habituellement un fichier, appelé le plus souvent, sur un système Unix, `/var/mail/$USER`[3], dans lequel les nouveaux messages sont délivrés. L'utilisateur peut lire les messages et les effacer ; s'il souhaite les classer dans des *dossiers* organisés par exemple par thème afin de pouvoir les retrouver rapidement, il devra utiliser son disque local (voir figure 5.2).

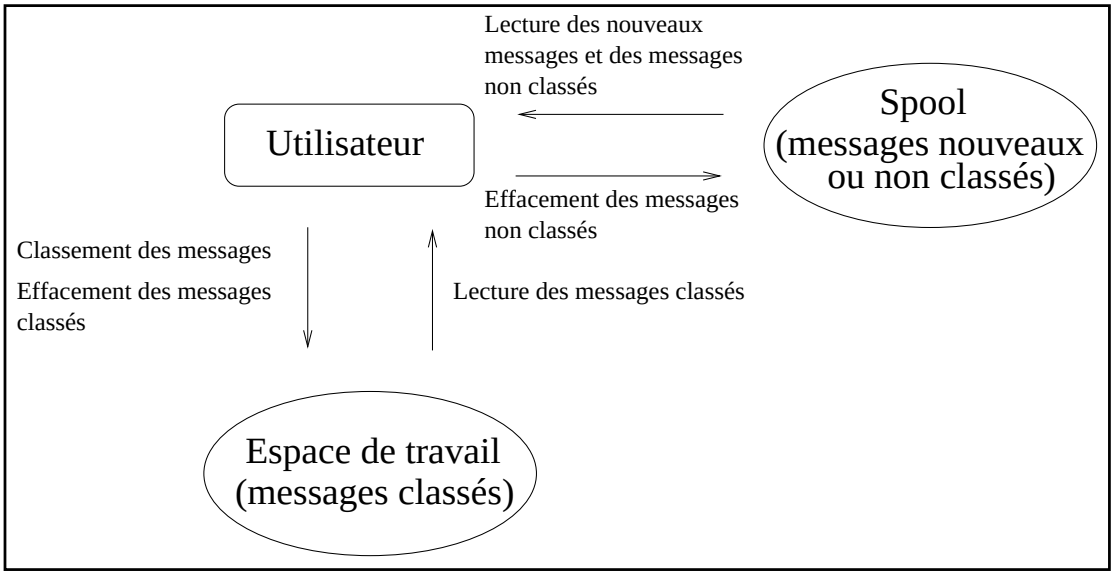

Figure 5.2 *Stockage et consultation par l'intermédiaire d'un spool*

3. `$USER` désigne ici le nom de connexion de l'utilisateur.

Ces dossiers sont conservés dans l'espace de travail privatif de l'utilisateur, c'est-à-dire que s'il souhaite les manipuler depuis un autre endroit que son environnement de travail habituel, il lui faudra s'y connecter afin d'effectuer les différentes opérations. Par contre, si l'utilisateur se trouve toujours sur le même poste de travail (ou un poste de travail partageant le même espace de fichiers), cette méthode est la plus simple à mettre en œuvre[4].

On peut rapprocher ce modèle de livraison du courrier du modèle traditionnel de la messagerie postale française : le courrier est délivré dans votre boîte aux lettres, vous pouvez l'y laisser après l'avoir lu (messages non classés), vous pouvez le jeter après l'avoir lu (ou même sans l'avoir lu) et vous pouvez le ranger chez vous dans des dossiers prévus à cet effet.

5.2.2 POP

La méthode POP (Post Office Protocol) se rapproche plus du modèle de la poste restante ; vous pouvez aller chercher vos messages lorsque vous le souhaitez, mais une fois que vous avez retiré vos nouveaux messages, vous ne pouvez pas les remettre dans votre boîte aux lettres[5].

Lorsque vous vous connectez à un serveur POP, votre ordinateur local rapatrie vos nouveaux messages (voir figure 5.3) et vous pouvez vous déconnecter du serveur pour pouvoir les lire et les manipuler (classement, destruction, etc.) *off-line*, en économisant notamment des frais éventuels de communication.

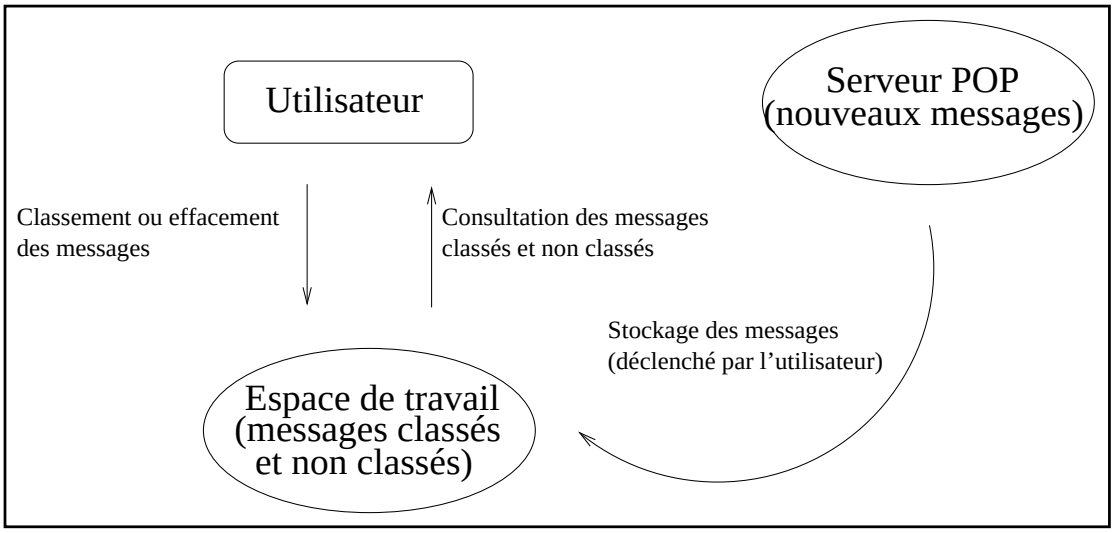

Figure 5.3 *Méthode POP*

Cette méthode convient parfaitement à des utilisateurs itinérants qui peuvent, en utilisant un

4. Cette méthode est celle utilisée par défaut par le programme standard de messagerie électronique, `sendmail`.

5. En fait, il est possible sur certains serveurs POP de ne pas effacer les messages lus, mais la plupart des clients POP gèrent incorrectement cette fonctionnalité et ne peuvent pas par exemple indiquer si un message est nouveau ou ancien mais non classé.

ordinateur portable et un modem, se connecter sur le site de leur entreprise régulièrement afin de rapatrier et envoyer leurs messages.

La section 5.5 page 207 explique comment configurer simplement un serveur POP pour une plate-forme Unix.

5.2.3 IMAP

Le modèle IMAP (Internet Mail Access Protocol) n'a pas d'équivalent immédiat dans le système de courrier postal français ; du point de vue électronique, c'est un modèle hybride entre le spool traditionnel (il faut être connecté pour lire les messages) et le modèle POP (on utilise un protocole et un serveur spécifiques pour y accéder et il est adapté à un certain type d'utilisateurs itinérants). Son utilisation est en train de devenir courante sur l'Internet, même s'il est pour le moment moins répandu que POP.

Le schéma fonctionnel (figure 5.4) est beaucoup plus simple que celui des deux autres méthodes et ne nécessite pas la présence d'un espace de travail fixé pour l'utilisateur. En effet, toutes les manipulations de messages (effacement, classement, etc.) sont effectuées par le serveur IMAP à la demande de l'utilisateur.

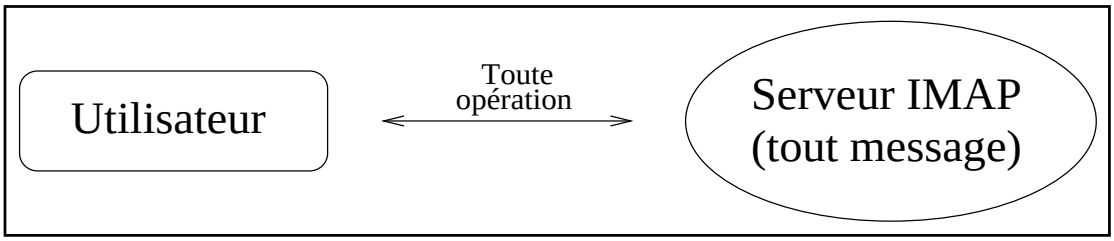

Figure 5.4 *Méthode IMAP*

Ce modèle est parfaitement adapté au cas de l'utilisateur itinérant qui n'utilise pas d'ordinateur portable mais qui se déplace de site en site, ces sites pouvant se connecter de manière interactive au serveur IMAP. L'utilisateur n'a besoin d'avoir avec lui en permanence que le programme client (par exemple sur disquette) lui permettant de manipuler son courrier sur le serveur.

Le protocole IMAP a de plus d'autres avantages sur le protocole POP, à savoir :

- Un serveur IMAP peut construire une liste des messages et la transmettre au client, alors qu'un client POP doit récupérer l'ensemble des messages afin de constituer lui-même cette liste ;
- le protocole MIME (voir page page 240) est complètement intégré à IMAP, ce qui signifie qu'un utilisateur peut, par exemple, demander à ne voir que les petites parties de texte d'un message sans avoir à en charger la totalité (il en découle une économie parfois énorme de bande passante, appréciée sur une ligne à bas débit).

5.2.4 Choisir une méthode

Comme on vient de le voir, il existe plusieurs méthodes adaptées à plusieurs situations qui peuvent être fondamentalement différentes. Le tableau 5.2 récapitule les différents choix possibles en fonction des besoins. Bien évidemment, il est possible de configurer simultanément plusieurs méthodes, afin de pouvoir accueillir un plus grand nombre de clients différents.

Utilisateur	**Possède un portable ?**	**Suggestion**	***Installation***
Fixe	—	spool	*très facile*
Itinérant	Oui	POP	*facile*
Itinérant	Non	IMAP	*moins facile*

Tableau 5.2 *Suggestion de méthode selon les cas de figure*

5.3 Les échangeurs de courrier électronique

5.3.1 Le transit du courrier

Il existe plusieurs types d'échangeurs de courrier électronique, aussi appelés MTA (Mail Transport Agent), et plusieurs modes d'échanges. Ceux-ci sont déterminés par les champs MX ou A du DNS (voir chapitre 4 page 145) ainsi que par des tables codées «en dur» dans les échangeurs autorisant un modèle statique d'échange de courrier.

Nous allons voir différentes manières de router un message à travers la matrice selon les configurations. Dans tous les cas, nous considèrerons que le client, c'est-à-dire le logiciel qui envoie le message à la demande de l'utilisateur, s'adresse à un serveur local qui délivrera le courrier électronique directement ou indirectement au serveur distant. Dans certains cas, le rôle du serveur local est joué par le client lui-même, notamment dans le cas où le client utilise directement le protocole SMTP (c'est le cas par exemple d'Eudora ou de Netscape).

Ces exemples ne sont présentés que pour illustrer le propos et ne prétendent en aucun cas constituer une liste exhaustive des possibilités existantes de routage du courrier électronique.

Accès direct

Le modèle à accès direct est le plus simple à comprendre (voir figure 5.5 page suivante). Le serveur local cherche dans ses tables internes s'il a le nom du serveur distant correspondant à l'adresse de destination indiquée, sinon, il cherche dans le DNS un enregistrement MX (Mail eXchanger) pour la machine cible ou, à défaut, un enregistrement A (adresse internet). Dès lors qu'il a une adresse où envoyer le message, il l'envoie (ou le garde pour l'envoyer plus

tard si la machine destinataire est indisponible) et, dans le cas contraire, génère un message d'erreur indiquant que la machine cible est inconnue.

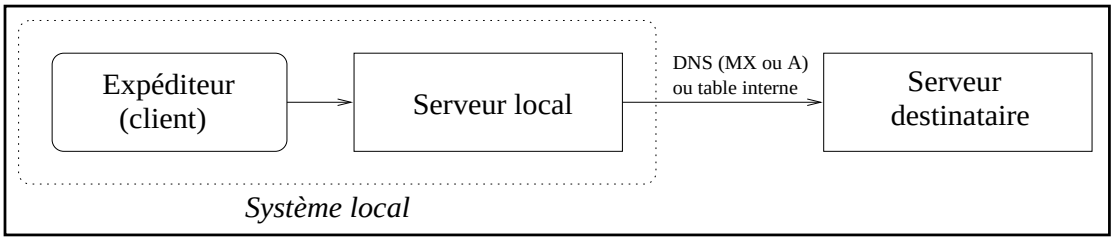

Figure 5.5 *Accès direct*

Chemin

Le plus souvent, une société ne souhaite pas que son serveur de messagerie central soit directement accessible depuis l'Internet (voir chapitre 14 page 427). Pour cela, les champs A et MX du DNS pointent sur un serveur (ou relais) exposé qui peut, quant à lui, directement accéder au serveur de messagerie destinataire (voir figure 5.6).

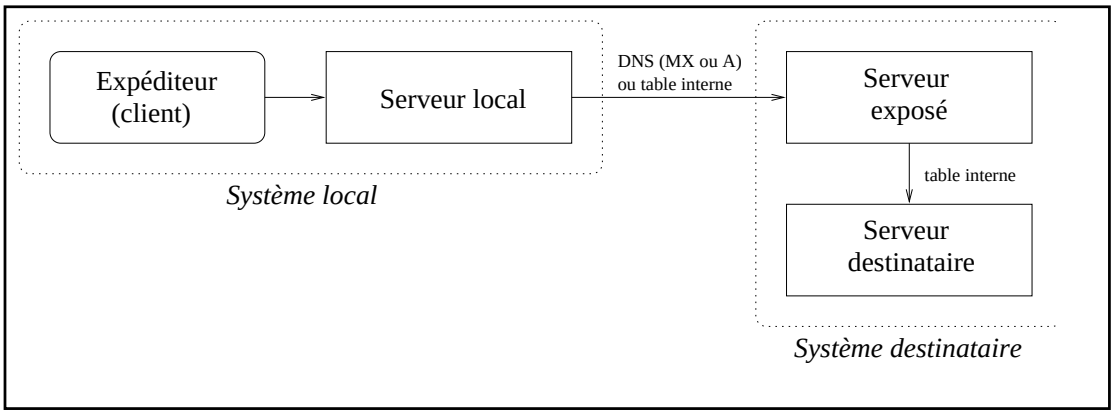

Figure 5.6 *Chemin obligatoire*

La configuration du serveur exposé est telle qu'il n'accepte que les courriers venant de l'extérieur et destinés à l'intérieur ou vice versa (il ne route pas par exemple les courriers à l'intérieur de la société). Cette configuration permet, en cas de problème de sécurité, de couper l'émission et la réception de courrier entre l'entreprise et le monde extérieur tout en ne perturbant pas les courriers internes.

Duplication

Il n'est pas toujours possible d'atteindre directement (en SMTP) une machine destinataire ou son relais officiel. Dans l'exemple de la figure 5.7 page suivante, le serveur local peut, s'il

choisit de ne pas suivre les champs A du DNS, préférer la route SMTP secondaire pour le destinataire qui est codée dans ses tables, à savoir la route vers la machine «relais 2».

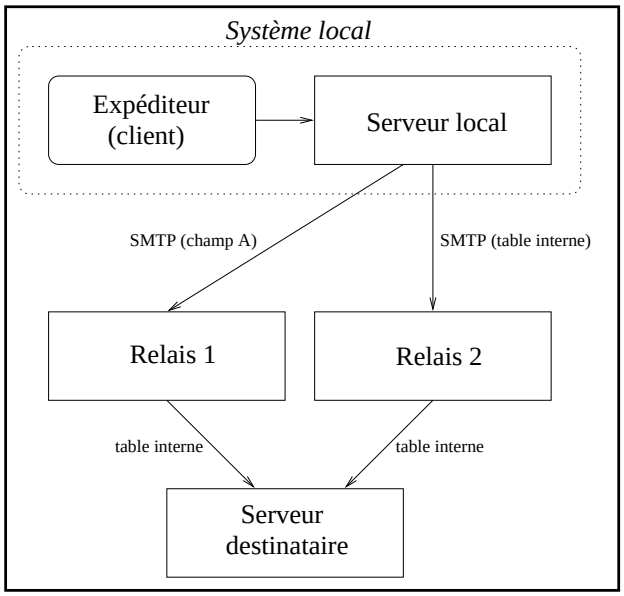

Figure 5.7 *Duplication des routes*

Ce relais enverra ensuite le message à la machine destinataire. Si l'expéditeur change la configuration de son serveur local, il est possible que les messages suivants passent par «relais 1» au lieu d'emprunter la route précédente. Le même scénario peut se produire si l'une des deux machines «relais 1» ou «relais 2» est inaccessible, ou encore si le serveur local, suivant la classification du message (urgent ou non urgent par exemple) décide de suivre la route directe, peut-être plus coûteuse financièrement ou la route indirecte qui utilise UUCP, plus économique dans certains cas.

5.4 Configuration d'un échangeur de courrier électronique : sendmail

Note préliminaire

Il est possible de trouver sur l'Internet des versions précompilées de sendmail. Ceci peut être un meilleur choix que de recompiler soi-même sendmail, notamment s'il manque des outils nécessaires à la compilation (comme gcc pour certaines plates-formes). Si vous souhaitez utiliser une version précompilée de sendmail, vous pouvez passer directement à la section 5.4.2 page 198 décrivant la configuration de sendmail.

Dans cette section, nous allons détailler l'installation et la configuration du programme **sendmail** pour plate-forme Unix. Cet échangeur de courrier électronique est, de loin, le plus répandu actuellement sur l'Internet.

5.4.1 Installation

Le programme `sendmail`, écrit par Eric ALLMAN, est disponible gratuitement à l'URL ci-dessous : `ftp://ftp.cs.berkeley.edu/pub/sendmail`. Le nom du fichier contenant l'archive est `sendmail.`*<numéro de version>*`.tar.gz` (par exemple, le nom de fichier `sendmail.8.8.3.tar.gz` sera utilisé pour la version 8.8.3).

Il faut ensuite décompresser et désarchiver le fichier et se rendre dans le sous-répertoire `src` de la distribution :

```
% gunzip -c sendmail.8.8.3.tar.gz | tar xf -
% cd sendmail-8.8.3/src
```

Si `sendmail` a été adapté à la plate-forme utilisée (ce qui est le cas de la plupart des systèmes Unix modernes), il suffit de taper la commande `makesendmail` qui se trouve dans le répertoire courant.

Ici, les exemples ont été réalisés sur Solaris 2.5.

```
% ./makesendmail
Configuration: os=SunOS, rel=5.5, rbase=5, arch=sun4, sfx=
Creating obj.SunOS.5.5.sun4 using Makefile.SunOS.5.5
Making dependencies in obj.SunOS.5.5.sun4
make: Nothing to be done for 'depend'.
Making in obj.SunOS.5.5.sun4
gcc -I. -O -I/usr/sww/include -DNDBM -DNIS -DNISPLUS -DSOLARIS=205
   -c alias.c -o alias.o
[...suite de la compilation...]
ld: fatal: library -l44bsd: not found
ld: fatal: File processing errors.  No output written to sendmail
make: *** [sendmail] Error 1
```

Si la compilation se termine sur une telle erreur, cela signifie que la nouvelle librairie BIND n'est pas installée sur le système. Le fichier `Makefiles/Makefile.SunOS.5.5` doit donc être édité afin d'enlever (aux alentours de la ligne 40) la référence à la librairie `44bsd` comme indiqué dans le fichier. Il faut ensuite relancer la compilation.

```
% ./makesendmail
Configuration: os=SunOS, rel=5.5, rbase=5, arch=sun4, sfx=
Making in obj.SunOS.5.5.sun4
gcc -o sendmail alias.o arpadate.o clock.o collect.o conf.o
   convtime.o daemon.o deliver.o domain.o envelope.o err.o
   headers.o macro.o main.o map.o mci.o mime.o parseaddr.o queue.o
   readcf.o recipient.o savemail.o srvrsmtp.o stab.o stats.o sysexits.o
   trace.o udb.o usersmtp.o util.o version.o  -L/usr/sww/lib -lresolv
   -lsocket -lnsl -lelf
groff -Tascii -mandoc aliases.5 > aliases.0
groff -Tascii -mandoc mailq.1 > mailq.0
groff -Tascii -mandoc newaliases.1 > newaliases.0
groff -Tascii -mandoc sendmail.8 > sendmail.0
```

Si le système n'avait pas eu le programme `groff` installé, la génération des pages de manuel aurait échoué. Dans ce cas, il aurait fallu rééditer le fichier `Makefile.SunOS.5.5` dans

le sous-répertoire `Makefiles` afin de demander à `make` d'utiliser `nroff -h` au lieu de `groff -Tascii` (comme indiqué dans le fichier lui-même).

On peut maintenant installer `sendmail`, en utilisant toujours le même script :

```
% ./makesendmail install
Configuration: os=SunOS, rel=5.5, rbase=5, arch=sun4, sfx=
Making in obj.SunOS.5.5.sun4
/usr/ucb/install -o root -g sys -m 6555 sendmail /usr/lib
for i in /usr/bin/newaliases /usr/bin/mailq; do \
   rm -f $i; ln -s /usr/lib/sendmail $i; done
/usr/ucb/install -c -o root -g sys -m 644 /dev/null \
    /var/log/sendmail.st
/usr/ucb/install -c -o root -g sys -m 444 sendmail.hf /etc/mail
```

Le nouveau `sendmail` est installé (le plus souvent dans le répertoire `/usr/lib`) avec des droits Unix corrects, ainsi que les différents liens (`mailq` et `newaliases`) dans `/usr/bin` et le fichier d'aide dans `/etc/mail`.

Attention

Le programme `sendmail` est installé, mais n'est pas encore prêt à fonctionner. Il vous faut maintenant créer un fichier de configuration ; cela fait l'objet des deux sections suivantes.

5.4.2 Principes de configuration de sendmail

Il y a des rumeurs qui ont la vie dure dans le monde de l'informatique, et des petites phrases qui circulent sans cesse depuis de nombreuses années. Parmi elles, on trouve :

« *Celui qui a configuré une fois `sendmail` connaît tout de l'informatique.* »

Au risque de décevoir le lecteur, il faut bien dire que ceci n'a plus rien de vrai. Autant la création d'un fichier `/etc/sendmail.cf` (nom du fichier de configuration standard de `sendmail`) était avant une véritable croisade contre la machine, autant maintenant, grâce au jeu de macro-commandes `m4`, c'est devenu un véritable jeu d'enfant.

En fait, la génération d'un fichier de configuration pour `sendmail` n'est ni plus ni moins que de la programmation (simplifiée à l'extrême, avouons-le). En effet, le fichier d'entrée (celui que nous avons à construire) peut être aussi simple que :

```
OSTYPE('solaris2')
FEATURE('nullclient', 'fenetre.fr')
```

Par contre, la sortie du programme m4 ressemble à :

```
R$* < $* > $* <@>    $: $1 < $2 > $3     unmark <addr>
R$* :: $* <@>        $: $1 :: $2         unmark node::addr
```

Quand on sait que ce sont de tels fichiers de configuration qu'il fallait avant produire à la main, on comprend mieux l'horrible réputation des fichiers de configuration de `sendmail`.

5.4.3 Création d'un fichier de configuration

Attention

Afin de configurer `sendmail`, il est conseillé (voire obligatoire sur certains systèmes) d'installer GNU `m4`, qui est une version gratuite de `m4`, les versions des constructeurs étant souvent problématiques. Cette version est disponible en France à l'URL `ftp://ftp.ibp.fr/pub/gnu`.

Quelques généralités concernant m4

m4 est un processeur de macro-commandes, c'est-à-dire qu'il considère l'entrée standard, et réagit selon le schéma suivant :

1. Si le texte en entrée commence par `dnl` (Delete until next New Line), alors le reste de la ligne est ignorée. Sinon, `m4` passe au point suivant.
2. Si le texte en entrée est à recopier sans modification (s'il est entouré par un *backquote* et un *quote*, comme dans `'ceci'`), alors il le recopie. Sinon, il passe au point suivant.
3. Si le texte en entrée est une commande connue, alors cette commande est exécutée (cela peut être soit une commande utilisateur soit une commande prédéfinie de `m4`) et son résultat réinterprété à partir du point 1, sinon il passe au point suivant.
4. L'entrée est recopiée telle quelle.

Par exemple, le fichier `test` dont le contenu suit :

```
abc(xxx)
‘define’
define(‘abc’, ‘le paramètre est $1’)
abc(xyz)
abc(abc(lll))
```

devient, à travers `m4` :

```
% m4 < test
abc(xxx)
define

le paramètre est xyz
le paramètre est le paramètre est lll
```

Comme on peut le remarquer, les fichiers d'entrée et de sortie présentent certaines similitudes, mais le fichier d'entrée a visiblement subi un traitement particulier. C'est justement le rôle de `m4` : appliquer certaines transformations à certaines parties d'un fichier.

Quelques explications sont nécessaires:

- `abc` n'est pas une commande m4, ni `xxx`. Ils sont donc recopiés tels quels ;
- `define` est un mot clé m4 mais on l'a protégé en écrivant `'define'`. Il est donc recopié tel quel ;
- la commande `define` définit une nouvelle macro-instruction `abc` qui affichera la chaîne `le paramètre est $1`, et remplacera `$1` par l'argument de la commande, puis repassera cette chaîne à travers m4 ; comme la définition de la commande se termine par un saut de ligne, une ligne blanche est affichée (voir ci-dessous pour apprendre comment enlever cette ligne vide) ;
- la commande `abc(xyz)` devient logiquement `le paramètre est xyz` ;
- la commande `abc(abc(lll))` est transformée en la chaîne de caractères `abc(le paramètre est lll)` qui devient donc, après une nouvelle phase d'expansion, `le paramètre est le paramètre est lll`

Comme nous l'avons vu, la définition d'une commande imprime une ligne vide car il y a un retour chariot à la fin de la ligne. Comme indiqué page précédente, la présence du mot-clé `dnl` permet de ne pas imprimer le reste de la ligne, notamment le retour chariot.

En ajoutant dans notre fichier `test` le mot-clé `dnl` à la fin de la ligne définissant la macro-instruction `abc` (juste après la parenthèse fermante, pour ne pas imprimer un espace), on obtient le résultat escompté :

```
% m4 < test
abc(xxx)
define
le paramètre est xyz
le paramètre est le paramètre est lll
```

Création du fichier de configuration

Les fichiers de configuration de `sendmail` se trouvent dans le répertoire `cf` de la distribution dont la structure est décrite figure 5.8. Ce répertoire comprend à son tour un certain nombre de sous-répertoires, dont le contenu est décrit dans le tableau 5.3 page suivante.

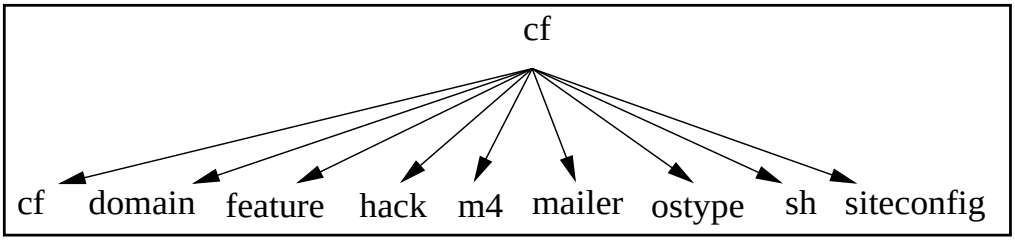

Figure 5.8 *Structure du répertoire `cf`*

Nous allons donc placer nos fichiers de configuration dans le répertoire `cf/cf`. Notre fichier aura un suffixe `.mc`, qui génèrera, après passage à travers m4, un fichier avec comme suffixe

`.cf`. C'est ce fichier qui sera ensuite installé à l'endroit où `sendmail` le cherche, à savoir `/etc/sendmail.cf`.

Répertoire	Contenu
`cf/cf`	Fichiers de configuration
`cf/domain`	Fichiers regroupant les caractéristiques d'un domaine
`cf/feature`	Possibilités offertes par `sendmail`
`cf/hack`	Possibilités supplémentaires
`cf/m4`	Fichiers internes (à ne surtout pas modifier)
`cf/mailer`	Définition des protocoles de transport à utiliser
`cf/ostype`	Caractéristiques des différents systèmes d'exploitation
`cf/siteconfig`	Tables internes aux sites (par exemple tables UUCP)

Tableau 5.3 *Sous-répertoires de `cf`*

Commandes de base

La première commande à inclure dans le fichier de configuration est celle qui permettra de charger le fichier contenant les définitions de base de `sendmail`:

```
include('../m4/cf.m4')
```

Toutes les autres commandes que nous allons utiliser maintenant auront été définies par ce fichier.

Ensuite, il nous faut chercher le fichier contenant les définitions propres à notre système d'exploitation. Ces fichiers se trouvent dans le répertoire `cf/ostype` de la distribution de `sendmail` et ont une extension `.m4`.

```
% ls ../ostype
aix3.m4          dynix3.2.m4      ptx2.m4
amdahl-uts.m4    hpux10.m4        riscos4.5.m4
aux.m4           hpux9.m4         sco3.2.m4
bsd4.3.m4        irix4.m4         solaris2.m4
bsd4.4.m4        irix5.m4         sunos3.5.m4
bsdi1.0.m4       isc4.1.m4        sunos4.1.m4
bsdi2.0.m4       linux.m4         svr4.m4
dgux.m4          nextstep.m4      ultrix4.m4
domainos.m4      osf1.m4          unknown.m4
```

Dans notre cas (Solaris), nous choisissons le fichier `solaris2.m4`. Il nous faut donc utiliser la commande :

```
OSTYPE('solaris2')
```

(la commande `OSTYPE`, comme beaucoup d'autres, a été définie lors de l'inclusion du fichier `cf.m4`).

Maintenant, nous allons pouvoir ajouter, une à une, les fonctionnalités souhaitées. La liste suivante n'est pas exhaustive mais devrait couvrir les besoins courants.

Choix des protocoles d'échange de messages

Il nous faut maintenant choisir les protocoles qui seront utilisés pour échanger les courriers électroniques avec les autres machines, ainsi que pour stocker (éventuellement) les messages entrants dans les boîtes aux lettres des utilisateurs.

Pour chaque protocole, il faudra ajouter une ligne dans le fichier de configuration de la forme `MAILER(nom)` ou `nom` désigne le nom du protocole. Par exemple, si nous souhaitons ajouter le protocole « local », il nous faudra inclure la ligne suivante dans le fichier de configuration :

```
MAILER('local')
```

(on peut ajouter `dnl` en fin de ligne pour éviter d'avoir une ligne blanche dans le fichier résultat).

Quelques-uns des protocoles présents sont :

local : ce protocole définit les *mailers* appelés « local » et « prog ». Ceux-ci sont tout le temps nécessaires, sauf dans le cas où tous les messages sans exception (y compris les messages d'erreur) sont renvoyés vers un autre site.

smtp : ce protocole définit les *mailers* appelés « smtp », « esmtp » (Extended Simple Mail Transport Protocol), « smtp8 » (extensions 8 bits pour SMTP) et « relay ».

uucp : ce protocole définit deux *mailers*, « uucp-old » (appelé également « uucp ») et « uucp-new » (« suucp »). Ils ne sont à inclure que si on souhaite pouvoir échanger du courrier à l'aide du protocole UUCP, ce qui est peu probable actuellement pour une entreprise.

usenet : ce protocole est un peu spécial : lorsqu'il est utilisé, il ajoute un ensemble de règles dans le fichier de configuration de `sendmail` autorisant l'envoi de messages à l'adresse électronique `group.usenet@notre.site`, sachant que ce courrier sera posté automatiquement dans le groupe de discussion « group » (voir chapitre 6 page 219 sur les forums de discussion).

fax : ce protocole permet, si vous installez le logiciel HylaFax (anciennement FlexFax) de Sam LEFFLER, d'envoyer des fax automatiquement lorsque vous envoyez un courrier électronique à un groupe d'adresses donné. Ce protocole a été développé à l'origine

pour pouvoir envoyer à faible coût des fax à l'intérieur des USA lorsque les communications locales sont payées au forfait.

procmail : ce protocole présente une interface avec le programme procmail. Utilisé avec l'option `FEATURE('local_procmail')` (voir ci-dessous), il vous permettra de bénéficier d'un système de livraison local plus puissant, plus flexible et plus sûr.

Choix des fonctions de sendmail

Les fonctions de `sendmail` s'utilisent sous deux formes différentes, avec ou sans deuxième argument. Par exemple, la fonction `use_cw_file` ne prend pas d'argument, alors que la fonction `mailertable` en prend un :

```
FEATURE('use_cw_file')
FEATURE('mailertable', 'dbm /usr/lib/mailertable')
```

Voici les fonctions suivantes susceptibles d'être utiles à notre configuration :

use_cw_file : récupère la liste des machines pour lesquelles `sendmail` va accepter du courrier dans le fichier `/etc/sendmail.cw`. Cela signifie que, par défaut, `sendmail` n'acceptera des messages que pour la machine elle-même, et pas pour les autres machines du domaine.

use_ct_file : récupère dans le fichier `/etc/sendmail.ct` la liste des utilisateurs que le programme `sendmail` doit autoriser à changer de nom (c'est-à-dire à envoyer un message depuis un autre nom sans rajouter de message d'avertissement). En général, ce sont les logiciels de listes de diffusion qui doivent figurer dans cette liste.

nouucp : permet de ne rien faire de spécial avec les adresses UUCP.

mailertable : permet d'utiliser la table indiquée en second argument (sous la forme de la chaîne « `dbm /usr/lib/mailertable` » par exemple) comme table statique indiquant les protocoles et arguments à utiliser pour un site ou une machine donnée.

domaintable : définit (en utilisant la même syntaxe que `mailertable`) une table de domaine, permettant notamment de gérer un changement de nom complet de votre domaine (en cas de changement de nom de la société par exemple) de manière transparente.

always_add_domain : ajoute le nom du domaine dans les messages échangés en interne. Si cette fonction n'est pas activée, un courrier adressé à `utilisateur` ne sera pas réécrit en `utilisateur@fenetre.fr`.

all_masquerade : permet, si la macro-instruction `MASQUERADE_AS` est appelée avec un nom en deuxième argument, de changer le nom de la machine interne d'où vient le courrier (ainsi que le nom des autres destinataires appartenant à la société) en un autre nom (en général, le nom de domaine de la société). Par exemple, les courriers envoyés par `utilisateur@machine.fenetre.fr` apparaîtront comme venant

de `utilisateur@fenetre.fr`. Des explications supplémentaires à ce sujet sont données ci-dessous.

nullclient : cette fonction, utilisée seule, suffit à renvoyer tous les courriers vers la machine dont le nom est donné en deuxième argument. Elle devra être utilisée sur tous les postes clients qui ont pour unique rôle de renvoyer les messages vers un agent de transport central.

local_procmail : permet d'utiliser le programme `procmail` comme délivreur de courrier local. Voir page précédente pour plus d'informations à ce sujet.

bestmx_is_local : permet de prendre en compte le courrier de toutes les machines dont l'enregistrement MX du DNS pointe sur la machine sur laquelle est lancé `sendmail` (voir chapitre 4 page 145 pour plus d'informations sur la configuration du DNS). Cela permet notamment de ne pas avoir à générer de fichier `sendmail.cw` contentant la liste des machines locales.

Prise en charge d'un domaine complet

Il est courant que, dans un domaine (par exemple `fenetre.fr`), une seule machine (dans notre cas `courrier.fenetre.fr`) soit chargée de délivrer les messages aussi bien localement que vers l'extérieur, toutes les autres machines s'adressant à celle-ci.

Afin de cacher le nom de la machine réelle d'où le courrier électronique est parti ainsi que le nom des machines que l'expéditeur aurait pu inclure par erreur en mettant des personnes de la société en copie, il faut utiliser les fonctions de masquage. Nous avons vu page précédente comment l'utilisation de la fonction `all_masquerade` permettait d'obtenir ce résultat, nous allons détailler ici un moyen de parachever l'œuvre.

Tout d'abord, nous utilisons la fonction `all_masquerade` ainsi que `MASQUERADE_AS` afin de cacher le nom de nos machines en `fenetre.fr`.

```
FEATURE('allmasquerade')
MASQUERADE_AS('fenetre.fr')
```

Par contre, la présence d'autres stations Unix peut poser des problèmes si par exemple les comptes `admin` et `daemon` envoient régulièrement des messages générés automatiquement : il peut être nécessaire de savoir de quelle machine viennent précisément ces messages. Pour cela, nous devons ajouter ces deux noms dans la liste des utilisateurs dits « exposés », c'est-à-dire que la fonction `all_masquerade` ne s'appliquera pas à ces adresses (en utilisant la commande `EXPOSED_USER` qui prend comme argument une liste d'utilisateurs séparés par des espaces) :

```
EXPOSED_USER('admin daemon')
```

De plus, on souhaite que les messages destinés au compte `root` soit délivrés localement (soit dans la boîte aux lettres de `root`, soit redirigés vers l'alias local). On rajoute donc la ligne:

```
LOCAL_USER('root')
```

dans le fichier de configuration.

Relais

Certaines variables peuvent être définies à l'aide de la commande `define` de m4 pour indiquer à `sendmail` les machines à utiliser comme relais. Un relais est un ordinateur auquel seront envoyés sans traitement tous les messages appartenant à une catégorie donnée. Ces variables et les types de messages concernés sont récapitulés dans le tableau 5.4.

Variable	Gère les messages avec comme destinataire ...
`LOCAL_RELAY`	... adresse sans nom de machine
`MAIL_HUB`	... adresse avec le nom de la machine locale
`SMART_HOST`	... adresse avec un autre nom de machine

Tableau 5.4 *Configuration des relais*

Noms d'utilisateurs

Un problème classique qui se pose dans une entreprise est « Comment faire pour que les messages semblent venir de l'adresse `Prenom.Nom@fenetre.fr` plutôt que de l'adresse `utilisateur@fenetre.fr` ? ». Une des réponses est l'utilisation des bases de données utilisateur.

Une base de données utilisateur est construite à partir du fichier texte `/etc/userdb` et génère un fichier `/etc/userdb.db` en utilisant la commande `makemap` fournie avec `sendmail`.

La compilation de `makemap` peut être problématique et n'est pas aussi facile que celle de `sendmail`.

Dans le cas de Solaris, il est possible de compiler `makemap` à partir de la séquence suivante :

```
% cd makemap
% make -f Makefile.dist CC="gcc -DSOLARIS" LIBS="-ldb"
```

La base de données utilisateur (format textuel) doit avoir la forme suivante :

```
utilisateur:mailname    Prenom.Nom
Prenom.Nom:maildrop     utilisateur
```

en remplaçant `Prenom`, `Nom` et `utilisateur` respectivement par le prénom, le nom et le nom de compte de chaque utilisateur. La base sera construite par la commande :

```
% makemap btree /etc/userdb.db < /etc/userdb
```

Ensuite, il faut indiquer à `sendmail` qu'il doit utiliser cette base de données. Il faut pour cela ajouter dans le fichier de configuration la ligne :

```
define('confUSERDB_SPEC', '/etc/userdb.db')
```

Exemple complet de fichier pour un serveur

Nous allons configurer un serveur pour le domaine `fenetre.fr` (les messages apparaîtront depuis ce nom), en acceptant les messages dirigés vers n'importe quelle machine de cette société (en supposant que le fichier `/etc/sendmail.cw` contient la liste des machines de la société). De plus, le nom du domaine devra être ajouté à tous les courriers, même internes.

Bien entendu, nous souhaitons que les courriers des utilisateurs apparaissent comme venant de `Prenom.Nom@fenetre.fr`, et nous avons construit les fichiers `/etc/userdb` comme indiqué page précédente.

De plus, nous voulons délivrer les courriers à l'aide du programme `procmail` que nous avons précédemment installé.

Le fichier de configuration est donc :

```
OSTYPE('solaris2')
MAILER('local')
MAILER('procmail')
MAILER('smtp')
FEATURE('allmasquerade')
FEATURE('use_cw_file')
FEATURE('always_add_domain')
FEATURE('local_procmail')
MASQUERADE_AS('fenetre.fr')
define('confUSERDB_SPEC', '/etc/userdb.db')
```

Exemple complet de fichier pour une station secondaire

Un serveur secondaire devra simplement renvoyer tout le courrier au serveur central de messagerie `courrier.fenetre.fr`.

Le fichier de configuration est donc simple :

```
OSTYPE('solaris2')
FEATURE('nullclient', 'courrier.fenetre.fr')
MASQUERADE_AS('fenetre.fr')
```

5.5 Configuration d'un serveur POP

Il existe trois versions du protocole POP : la première, qui n'est plus du tout utilisée, la deuxième (POP2) que l'on trouve encore sur certains systèmes et la troisième (POP3) qui est actuellement la plus répandue sur l'Internet.

Le tableau 5.5 présente différents serveurs POP3 présents sur l'Internet.

Plate-forme	Serveur POP3 proposé	Disponibilité
Machines Unix	Popper version 2.2	`ftp://ftp.qualcomm.com/`
DEC VAX	IUPOP3	`ftp://ftp.indiana.edu/`
IBM (mainframe)	POPD	`ftp://vmd.cso.uiud.edu/`
Windows 3.1	SLmail16 (commercial)	`http://www.seattlelab.com/`
Windows 95	SLmail95 (commercial)	`http://www.seattlelab.com/`
Windows NT	SLmailNT (commercial)	`http://www.seattlelab.com/`
Novell	Mercury	`ftp://ftp.qualcomm.com/`
Macintosh	Apple Internet Mail Server	`http://www.solutions.apple.com/`

Tableau 5.5 *Différents serveurs POP3*

La procédure d'installation d'un serveur POP3 est très simple. Nous allons la détailler pour le serveur `qpopper` sur plate-forme Solaris.

Tout d'abord, il faut récupérer le fichier contenant `qpopper` par FTP sur le serveur de Qualcomm[6], le décompresser et le désarchiver:

```
% zcat qpop2.2.tar.Z | tar xf -
% cd qpopper2.2
```

Ensuite, il est conseillé d'éditer le fichier `make` correspondant à la plate-forme désirée (dans notre cas `make.solaris2`) ainsi que le fichier `popper.h` afin d'y ajuster certaines variables si nécessaires ; on peut notamment souhaiter utiliser le compilateur `cc` de Sun au lieu de `gcc`, qui est le compilateur choisi par défaut.

Pour cela, il faut, dans le fichier `make.solaris2`, mettre en commentaire la ligne définissant `gcc` comme compilateur et la remplacer par une ligne définissant `cc` :

6. `ftp://ftp.qualcomm.com/quest/unix/servers/popper/qpop2.2.tar.Z`

```
#CC = gcc -g -fstrength-reduce -fpcc-struct-return
CC = cc
```

Pour construire le serveur, il suffit maintenant d'appeler `make` avec comme argument le nom du système :

```
% make solaris2
cc -O -DSOLARIS2 -DSYSV -DBIND43 -DHAVE_VSPRINTF -DAUTH -DMAILOCK
  -DDEBUG -DBINMAIL_IS_SETGID -DCONTENT_LENGTH=1 -DGNU_PASS
  -DNO_GETLINE -DAPOP=\"/etc/pop.auth\" -DPOPUID=\"pop\"    -c flock.c
  -o flock.o
[...]
cc flock.o pop_dele.o pop_dropcopy.o pop_get_command.o
  pop_get_subcommand.o pop_init.o pop_last.o pop_list.o pop_log.o
  pop_lower.o pop_msg.o pop_parse.o pop_pass.o pop_quit.o pop_rset.o
  pop_send.o pop_stat.o pop_updt.o pop_user.o pop_xtnd.o pop_xmit.o
  popper.o pop_bull.o xtnd_xlst.o pop_uidl.o mktemp.o pop_rpop.o
  pop_apop.o md5.o -o popper.solaris2 -lsocket -lnsl -lresolv -lkrb
  -lmail
```

Le serveur est maintenant prêt à être installé. On peut, par exemple, l'installer dans le répertoire `/usr/local/lib`, qui est le plus souvent utilisé.

```
% /usr/ucb/install -m 755 -o root -g other popper /usr/local/lib
```

Une fois que cela est fait, la mise en service se fait par l'ajout d'une ligne dans le fichier `/etc/inet/inetd.conf`[7], afin d'indiquer au système que le serveur doit être lancé lorsqu'une requête est faite sur le port correspondant :

```
pop3 stream tcp nowait root /usr/local/lib/popper popper -s
```

Si le fichier de définition de services `/etc/services` ne contient pas le numéro de port associé au service POP3, il suffit de l'ajouter :

```
pop3    110/tcp    # Post office protocol
```

Il faut maintenant envoyer au programme `inet` le signal `SIGHUP` afin qu'il relise son fichier de configuration :

```
% /usr/ucb/ps aux | grep inet | grep -v grep
root   124  0.0  3.9  804  276  ?  S   00:47  0:00 /usr/sbin/inetd
% kill -HUP 124
```

Le nouveau serveur est maintenant fonctionnel. Dès maintenant, les utilisateurs peuvent configurer leurs logiciels lecteurs de courrier électronique pour qu'ils pointent sur ce serveur.

7. `/etc/inetd.conf` sur certains systèmes

5.6 Configuration des postes clients

5.6.1 Netscape

L'utilisation de Netscape comme lecteur de courrier électronique simplifie grandement la tâche de l'administrateur système, qui n'a à expliquer à ses utilisateurs que la configuration d'un seul produit, puisque `Netscape` est disponible à la fois sur les machines Unix, les machines Windows (3.1, 95 et NT) et les Macintosh.

La configuration se fait à travers l'interface graphique de Netscape, par le choix « *Mail and News preferences...* » du menu « *Options* ». Il faut procéder en cinq étapes, que nous allons détailler ci-dessous.

L'onglet « *Appearance* »

Les choix disponibles dans cet onglet sont présentés figure 5.9 page suivante. L'utilisateur, à partir de ce formulaire, peut configurer les paramètres suivants :

Apparence du message : vous pouvez choisir d'afficher le message en police fixe (police courier par exemple) ou en police proportionnelle. Le second choix est plus agréable à regarder, mais il risque de casser les alignements volontaires qui auraient pu être réalisés par l'auteur du message.

Apparence du texte cité : lorsqu'un auteur de courrier reprend dans son message une partie du texte auquel il répond, Netscape permet de faire apparaître ce texte d'une manière différente afin de pouvoir facilement le distinguer de la réponse elle-même. Par exemple, sur la figure, on a choisi de mettre le texte en italique.

Organisation des fenêtres : les différentes fenêtres de la messagerie de Netscape peuvent être alignées horizontalement, verticalement ou affichées les unes sur les autres. Dans l'exemple, les fenêtres sont toutes alignées sur l'horizontale.

L'onglet « *Compose* »

Dans cet onglet, l'utilisateur peut configurer les paramètres qui seront utilisés lorsqu'il composera un nouveau message ou une réponse à un message existant. La figure 5.10 page suivante montre une configuration possible de ces paramètres.

L'utilisateur peut donc configurer les paramètres suivants :

Type de message : lorsqu'un auteur de message souhaite utiliser les caractères accentués français (ou tout autre caractère ne faisant pas partie du jeu de caractères américain ASCII), il se peut que le message ait besoin d'être recodé dans un jeu de caractères moins étendu afin d'être sûr qu'aucune passerelle de messagerie n'effacera les caractères qu'elle pourrait juger illicites, c'est le rôle de l'option « *Quoted Printable* ».

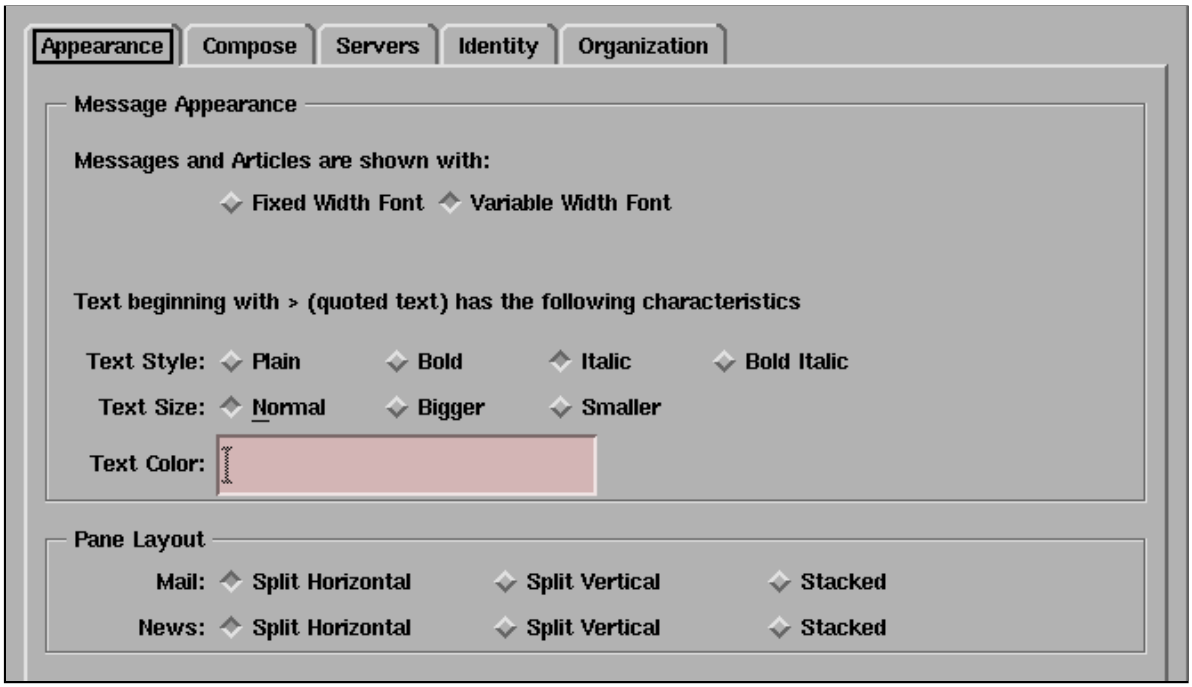

Figure 5.9 *Saisie des paramètres de présentation des messages*

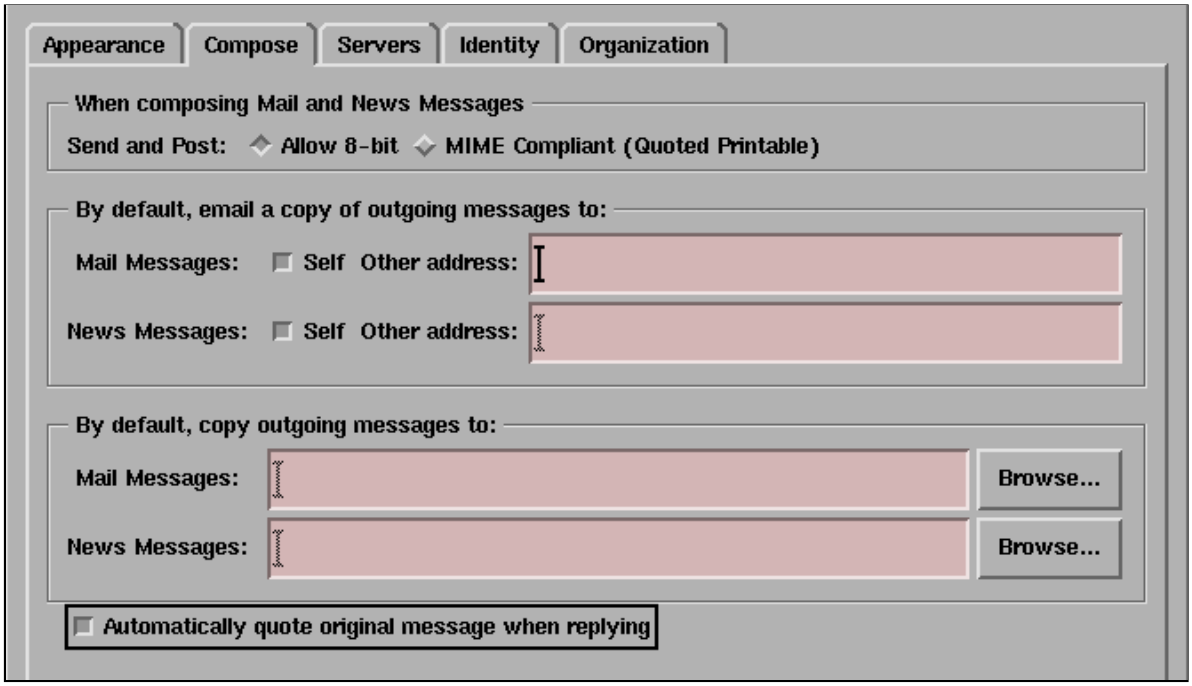

Figure 5.10 *Saisie des paramètres de composition*

Cependant, aujourd'hui tous les nouveaux systèmes de messagerie installés autorisent les jeux de caractères étendus, c'est pourquoi il est conseillé de choisir l'option « *Allow 8 bit* ». On peut également noter que l'utilisation du format MIME dans les groupes de diffusion est en général proscrit.

Copie des messages : l'utilisateur peut, s'il le désire, configurer Netscape pour qu'il lui envoie une copie de ses propres messages, à des fins d'archivage. Il peut même envoyer systématiquement une copie à une autre adresse, par exemple le responsable des relations publiques de l'entreprise qui stocke l'ensemble des messages envoyés par les commerciaux.

Citation automatique : lorsqu'un utilisateur a l'habitude de citer une partie du message auquel il répond, il peut être préférable pour lui de choisir l'option de citation automatique ; en effet, dans certains cas, il est plus rapide d'effacer une citation qu'on ne souhaite pas inclure plutôt que de cliquer sur l'option correspondante dans Netscape pour inclure explicitement une copie lorsqu'on en a besoin.

L'onglet « *Servers* »

Cet onglet permet de configurer les adresses des différents serveurs utilisés pour lire ou envoyer des messages (voir figure 5.11 page suivante). Il est très important de définir correctement ces paramètres sous peine de ne pas pouvoir lire ou écrire de messages.

Serveur SMTP : ce champ doit contenir l'adresse d'un serveur SMTP, par exemple l'adresse d'une machine sur laquelle tourne le programme `sendmail` (voir section 5.4 page 196 pour la configuration d'un tel serveur). Ce serveur doit être capable, directement ou indirectement, d'envoyer un message vers un utilisateur se trouvant dans la matrice.

Serveur POP3 : dans cet exemple, on a choisi d'utiliser un serveur POP3 pour lire le courrier électronique (la section 5.5 page 207 indique comment installer un tel serveur). Il faut indiquer le nom de la machine sur laquelle tourne ce serveur, ainsi que le nom de la boîte aux lettres dans laquelle les courriers destinés à l'utilisateur sont délivrés. Attention : en général, ce n'est pas l'adresse sous la forme `Prénom.Nom` de l'utilisateur, mais son nom de compte Unix composé en général d'au plus huit symboles.

Taille maximum des messages : l'utilisateur peut souhaiter ne pas recevoir de messages de taille trop importante, notamment s'il est connecté par modem par le réseau téléphonique commuté. Dans ce cas, il peut choisir une taille maximum de message accepté, les messages trop gros étant ignorés.

Nettoyage de la boîte aux lettres : il est possible de laisser les messages sur le serveur et de ne pas les effacer même une fois qu'ils ont été rapatrié en local ; ceci n'est pas conseillé pour des raisons évidentes d'encombrement de boîte aux lettres sur le serveur.

Présence de nouveaux messages : Netscape peut, si l'utilisateur le souhaite, vérifier toutes les n minutes si un nouveau message est arrivé dans la boîte aux lettres de l'utilisateur et l'en avertir.

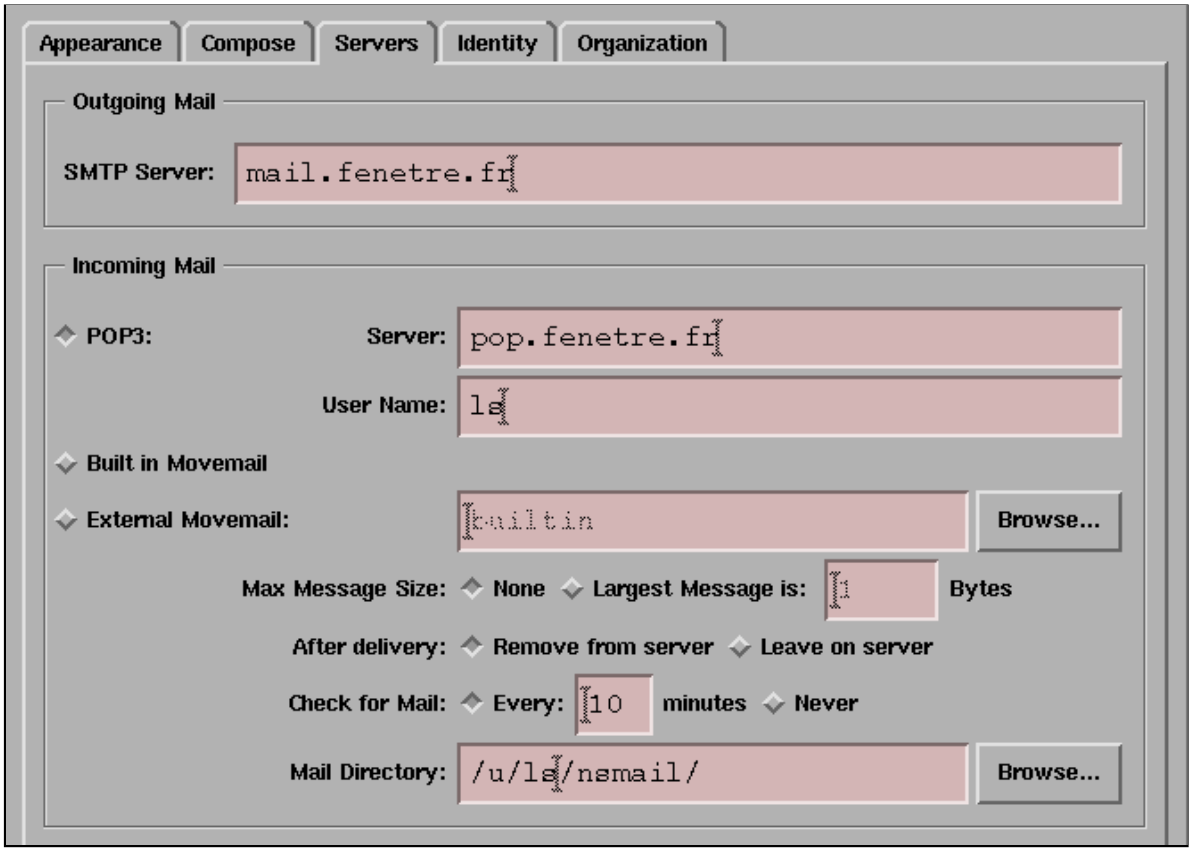

Figure 5.11 *Saisie de la configuration des serveurs*

Répertoire de stockage : ce répertoire contiendra, à terme, tous les messages que l'utilisateur aura rapatriés depuis son serveur POP3.

L'onglet « *Identity* »

Cet onglet affiche une fenêtre présentée figure 5.12 page ci-contre à partir de laquelle l'utilisateur peut configurer ses paramètres personnels, permettant aux destinataires de ses messages de lui répondre ou de savoir à quelle organisation il appartient.

Nom : c'est le nom complet de l'utilisateur (ici Luc STONED) tel qu'il apparaîtra dans le message.

Adresse électronique : cela peut être n'importe quelle adresse valable pour l'expéditeur. Par exemple, Luc STONED aurait pu choisir l'adresse électronique `ls@fenetre.fr` (`ls` étant son nom de compte Unix et le nom de sa boîte aux lettres), mais il a préféré `luc.stoned@fenetre.fr`, le trouvant plus explicite et plus facile à retenir pour les destinataires de ses messages.

Organisation : cette boîte doit contenir le nom de la société. Il apparaîtra dans le champ « *Organization* » des courriers envoyés.

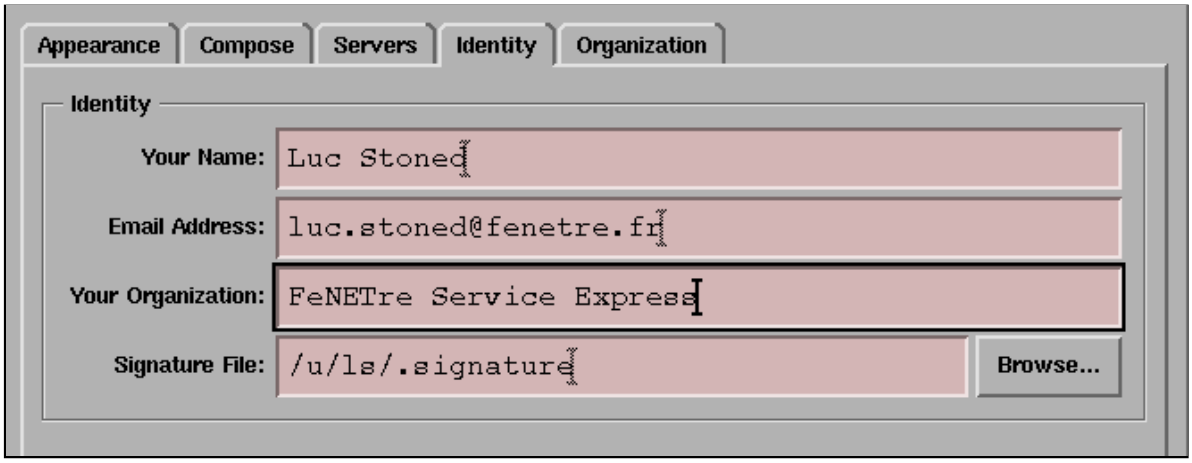

Figure 5.12 *Saisie des paramètres utilisateur*

Fichier de signature : c'est un nom de fichier qui contient habituellement les coordonnées professionnelles de l'utilisateur, quelquefois associées à une petite phrase philosophique ou humoristique. Attention toutefois à ne pas mettre un texte trop long sous peine de se le voir reprocher (voir section 5.7 à ce sujet).

L'onglet « *Organization* »

Contrairement à ce que le nom pourrait laisser croire, la fenêtre affichée lorsque l'on sélectionne cet onglet n'a rien à voir avec l'entreprise dans laquelle l'utilisateur se trouve ; elle concerne l'organisation et la présentation des messages de l'utilisateur. Un exemple de configuration est présenté figure 5.13 page suivante.

Retenir le mot de passe : lorsque cette option est activée, Netscape enregistre le mot de passe de l'utilisateur pour ne pas le lui redemander lors des connexions futures. Il faut savoir que cette option est fort pratique lorsque l'utilisateur est la seule personne à utiliser la machine sur laquelle Netscape fonctionne, mais également très dangereuse car un utilisateur malicieux n'aura aucune difficulté à aller lire ce mot de passe et à l'utiliser plus tard à des fins plus ou moins avouables.

Présentation sous forme d'arbre : cette option permet de présenter les courriers successifs sous une forme hiérarchisée plutôt qu'à la suite les uns des autres. Cela permet, lorsqu'on est abonné à des listes de diffusion, de suivre plus facilement le cours d'une discussion à laquelle participent de nombreux protagonistes.

Tri des messages : les messages peuvent être triés selon différents critères qui permettent par exemple de retrouver rapidement tous les messages venant d'une personne donnée. Le choix par défaut, qui est le tri par date d'arrivée, est le plus intuitif, et permet de suivre rapidement l'évolution d'une discussion ou de reconstruire l'historique d'une situation.

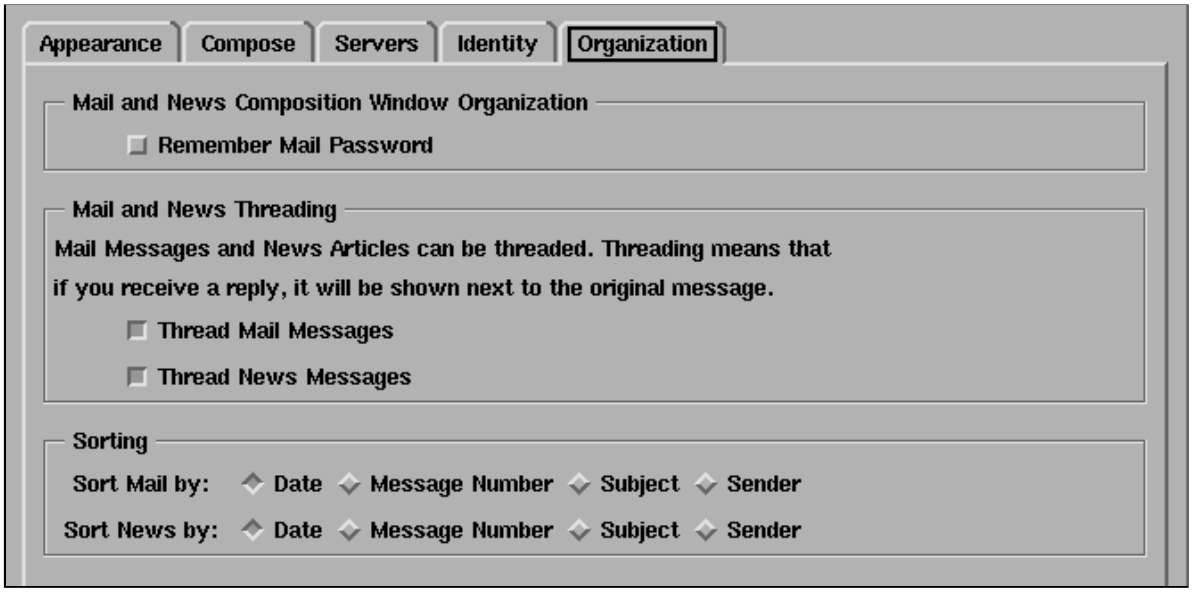

Figure 5.13 *Saisie des paramètres d'organisation*

5.6.2 ELM

Le programme `ELM` est un client permettant de lire et d'envoyer des messages en mode terminal plein écran, c'est-à-dire qu'il est utilisable aussi bien dans une fenêtre X-Window qu'à distance par modem. Ceci en fait le client favori des utilisateurs d'Unix.

Installation

`ELM` est disponible à l'URL `ftp://ftp.ibp.fr/pub/unix/mail/elm/`. L'installation se fait en suivant la procédure suivante :

```
% mkdir elm
% cd elm
% gunzip -c ../elm2.4.tar.gz | tar xpBf -
% ./Configure

Beginning of configuration questions for elm2 kit.

First let's make sure your kit is complete.  Checking...
Looks good...
Making bin directory

Checking your sh to see if it knows about # comments...
Your sh handles # comments correctly.
[...]
```

Le programme d'installation d'`ELM` demande ensuite à l'administrateur d'indiquer ses choix, à travers une série de questions. Le tableau 5.6 page ci-contre décrit les principales parties configurables du système.

Option	Valeur conseillée
Utilisation du calendrier	L'activation de cette option permet à `ELM` de guetter dans les messages reçus d'éventuelles commandes destinées à l'agenda électronique du destinataire.
`isprint()` fonctionne pour des caractères 8 bits	Il est conseillé de répondre par l'affirmative si l'on ne sait pas, et de reconfigurer ensuite `ELM` si certains caractères sont mal affichés.
Nom complet de l'utilisateur	Normalement, le prénom et le nom de l'utilisateur se trouvent dans le champ `GCOS` du fichier `/etc/passwd`.
Programme de mise en page pour la documentation	Il est conseillé d'utiliser `groff` si celui-ci est disponible, `troff` sinon.
Modèles de mémoire	Répondre `none` est conseillé.
Compilateur C	`gcc` s'il est disponible, `cc` sinon.
Bibliothèques additionnelles	Laisser les valeurs proposées par le programme de configuration.
Chemin du programme délivrant le courrier	`/usr/lib/sendmail`.
Fonction d'édition de boîte aux lettres	Il est conseillé, s'il est prévu que des utilisateurs inexpérimentés utilisent `ELM`, de ne pas autoriser cette option qui peut endommager la boîte aux lettres.
Support MIME	L'autoriser uniquement si le programme `metamail` a été installé.
Endroit où placer les exécutables	`/usr/local/bin` est traditionnellement choisi.

Tableau 5.6 *Options de configuration d'`ELM`*

Utilisation

`ELM` se lance par la commande `elm`, avec, éventuellement, un nom de fichier contenant une boîte aux lettres précédé de `-f`. Après avoir proposé de créer un répertoire dans l'espace privatif de l'utilisateur destiné à accueillir les dossiers contenant les messages classés, `ELM` ouvre une fenêtre en mode plein écran telle que celle présentée figure 5.14 page suivante.

La lecture et le défilement d'un message se font simplement en appuyant sur la barre d'espace. Le fait d'appuyer deux fois sur la touche « `?` » permet d'obtenir de l'aide sur les touches utilisables depuis `ELM`.

L'envoi de message se fait en appuyant sur la touche « `m` » et en suivant les instructions affichées. L'utilisation d'`ELM` est en fait suffisamment simple pour ne pas être détaillée ici. Notons simplement qu'en appuyant sur la touche « `o` » à partir du menu principal, on peut configurer un certain nombre d'options d'`ELM` (voir figure 5.15 page suivante).

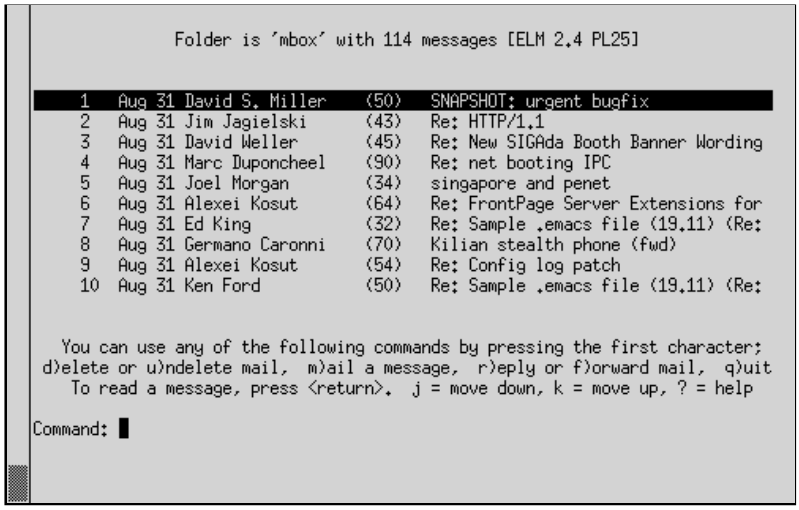

Figure 5.14 *Fenêtre d'accueil d'ELM*

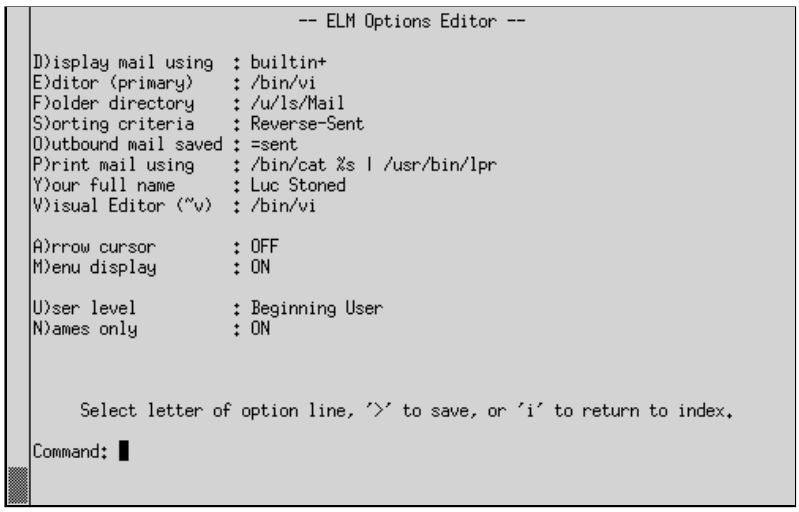

Figure 5.15 *Options d'ELM*

5.7 Notions de nétiquette

À ses débuts, l'Internet était un lieu de rencontre privilégié de chercheurs, d'universitaires et d'étudiants qui étaient animés par le même désir de découverte. De plus en plus, la population de l'Internet tend à représenter la population du monde réel, permettant ainsi à un grand nombre de personnes d'utiliser les nouvelles technologies.

Autant les premiers utilisateurs étaient empreints d'une culture qui se transmettait facilement de bouche à oreille (il faudrait écrire « de clavier à écran »), autant l'arrivée massive de nouveaux individus nécessite une éducation conséquente afin de rappeler les us et coutumes qui ont fait de l'Internet un réseau convivial et utile. Il existe un RFC (le 1855) intitulé « *Netiquette Guidelines* ». On désigne en effet par « nétiquette » l'étiquette à respecter sur l'Internet.

Nous en rappelons ici les principes généraux, dont tous les nouveaux utilisateurs devraient prendre connaissance.

5.7.1 Communication privée

Les principes généraux à suivre pour la communication privée (c'est-à-dire notamment le courrier électronique) sont les suivants :

- Le courrier sur l'Internet n'est absolument pas sécurisé ; dès lors qu'un message peut sortir de votre réseau, il est dangereux d'y écrire des choses que vous ne mettriez pas sur une carte postale sans enveloppe.
- Il ne faut jamais répondre aux « lettres en chaîne » qui polluent l'Internet. Si vous recevez une telle lettre et que l'on vous demande de la renvoyer à 5 autres personnes, ne le faites pas et prévenez votre administrateur système, qui prendra les mesures nécessaires.
- La culture et les mœurs du destinataire de votre message vous sont peut-être inconnus ; ne prenez pas de risque en envoyant un message dont l'humour ou la légèreté sont susceptibles de choquer celui qui le reçoit.
- Inversement, acceptez le fait que d'autres aient un humour différent du vôtre, et ne vous offusquez pas trop facilement. Le mieux est d'attendre le lendemain avant de répondre à un courrier de manière trop émotionnelle.
- Gardez une trace des courriers que vous envoyez. Si un courrier de 500 lignes qui vous a pris la nuit à saisir est perdu à cause d'une erreur dans l'adresse du destinataire[8], même votre administrateur système n'y pourra rien.
- Il ne faut pas abuser des « *smileys*[9] », symboles permettant de rendre les messages plus expressifs (visage souriant, visage pleurant, ...). Les caractères « : -) » représentent par exemple, si vous tournez le livre d'un quart de tour sur la droite, une figure en train de sourire. La présence de ce symbole ne vous autorise pas à dire n'importe quoi.
- Évitez les lignes de plus de 72 caractères qui ne seront probablement pas présentées correctement lorsque quelqu'un répond à l'un de vos messages.
- Gardez votre signature relativement courte. Il est couramment admis qu'une signature de 4 lignes est un maximum.

5.7.2 Communication de groupe

Toutes les règles de la communication personnelle s'appliquent également à la communication de groupe, que ce soit dans les listes de diffusion ou les forums. On peut y adjoindre les

8. Habituellement, le programme qui envoie les messages vous renvoie les courriers électroniques avec une adresse de destination invalide, mais ce n'est pas une règle absolue.

9. Appelés aussi « sourillards » ou « emoticons » par nos amis Québécois.

suivantes :

- Suivez les discussions sur les groupes que vous venez de rejoindre pendant un minimum de temps afin de savoir quels us et coutumes les régissent.
- Méfiez-vous de ce que vous écrivez, encore plus que pour les messages personnels. Vous risquez de choquer un plus grand nombre de personnes.
- Votre patron, présent ou futur, peut très bien lire ou récupérer les archives des groupes auxquels vous participez, donc faites attention à ce que vous postez.
- Ne répondez à tout le groupe que lorsque votre contribution peut apporter quelque chose ; n'envoyez pas non plus de messages disant seulement « Moi aussi ».
- Lorsque vous n'appliquez pas la règle précédente par erreur, envoyez immédiatement un message d'excuse au groupe.
- Pour les listes de diffusion, gardez une trace du message d'accueil indiquant comment vous désabonner.

6

Les forums de discussion

Les séances de travail collectif sur l'Internet peuvent se dérouler de plusieurs façons ; parmi elles, on trouve les listes de diffusion, les conférences en temps réel (voir le chapitre sur le multicast page 379) ou les forums. Ces derniers font l'objet du présent chapitre ; ils sont largement utilisés sur l'Internet, car, contrairement aux listes de diffusion, ils ne nécessitent pas de procédure complexe d'abonnement ou de désabonnement et ne risquent pas de saturer une boîte aux lettres pendant les congés de l'utilisateur.

6.1 La notion de forum

Les forums de discussion, aussi appelés « *Usenet news* », traitent de sujets aussi divers que le langage de programmation Ada ou la cuisine à l'ancienne, en passant par la littérature française ou le droit à la vie privée dans les pays nordiques.

Il existe plusieurs dizaines de milliers de **groupes** et il est parfois difficile de ne pas se noyer dans l'énorme liste que cela implique.

6.1.1 L'abonnement à un groupe

Contrairement aux listes de diffusion, il n'y a pas besoin de s'abonner réellement à un groupe afin de pouvoir le lire ou y écrire. Le schéma de la figure 6.1 page suivante décrit la différence entre les deux systèmes : alors que la personne abonnée à une liste de diffusion reçoit par courrier électronique tous les messages envoyés à cette liste, le lecteur d'un groupe doit effectuer la démarche consistant à aller lire les messages arrivés depuis la dernière fois qu'il a consulté ce groupe.

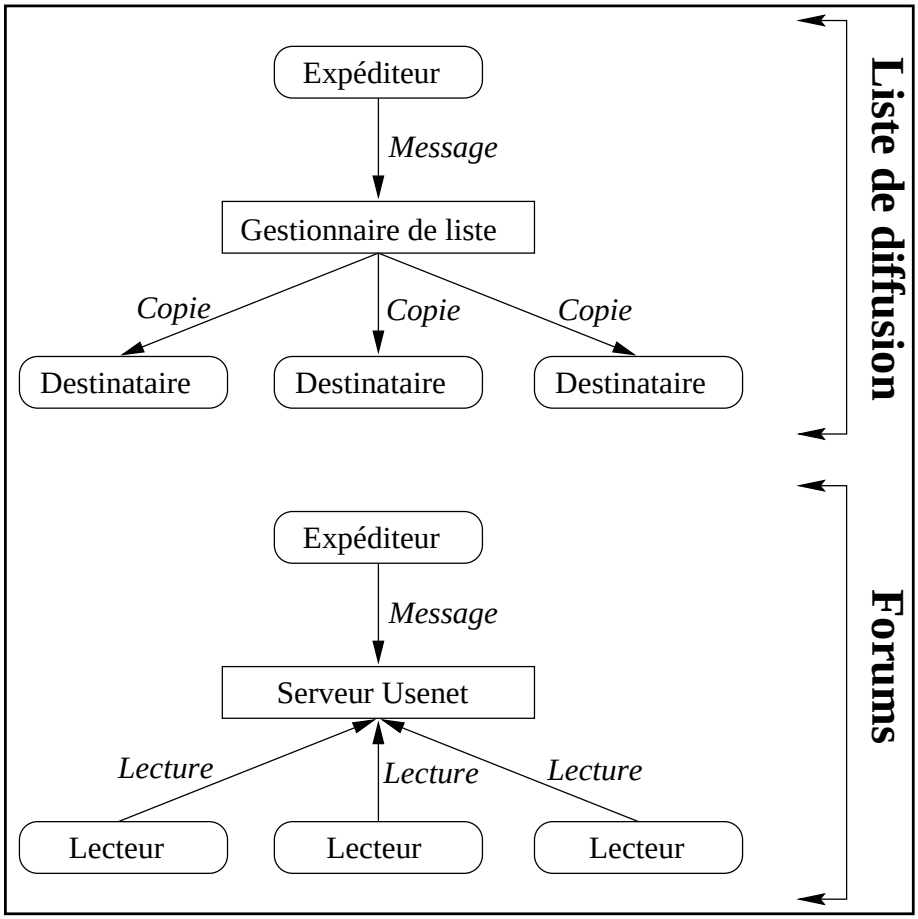

Figure 6.1 *Listes de diffusion et forums*

Chacun des deux systèmes possède ses avantages propres :

- les listes de diffusion permettent une meilleure réactivité au sens où les abonnés sont prévenus à chaque fois qu'un participant poste un message ;
- les forums de discussion permettent de ne participer au débat que lorsqu'on le souhaite et ne surchargent pas les boîtes aux lettres lorsqu'on n'est pas joignable pendant un temps relativement long.

Le logiciel permettant de lire les forums garde habituellement une liste des messages que l'utilisateur a déjà lus dans le fichier `newsrc@.newsrc`, ce qui lui permet, lorsque l'utilisateur se connecte à nouveau au serveur, de ne lui présenter que les messages arrivés depuis sa dernière visite.

6.1.2 Poster dans un groupe

Lorsque l'on souhaite participer à une discussion, il suffit de poster un message à destination du groupe. Ce message pourra, à son tour, être lu par toutes les personnes souhaitant consulter le groupe à partir de ce moment-là.

Puisqu'il n'y a pas de liste de personnes abonnées à un groupe, n'importe qui peut, *a priori*, poster dans n'importe quel groupe. Ceci explique pourquoi on trouve actuellement tant de messages publicitaires envoyés plusieurs fois dans tous les groupes existants sans considération aucune pour les destinataires : ces messages sont destinés à avoir la plus grande audience possible, et les émetteurs ne se soucient guère de gêner les autres. De même, on trouve souvent des messages qui n'ont rien à faire dans un groupe donné, à cause de la méconnaissance des contenus des groupes de la part d'utilisateurs souvent débutants. Pour ces deux raisons, on peut apprécier l'existence de groupes dits « modérés », qui permettent d'éliminer en partie ces problèmes.

6.1.3 Les groupes modérés

Certains groupes n'acceptent pas les articles en provenance de personnes non identifiées : lorsque quelqu'un souhaite poster une contribution, il envoie un courrier électronique au **modérateur** qui, lui, est autorisé à renvoyer l'article dans le forum. En fait, le courrier électronique est envoyé automatiquement dès lors qu'une personne non habilitée à poster dans le groupe essaye de le faire sans passer par le modérateur, comme indiqué sur la figure 6.2.

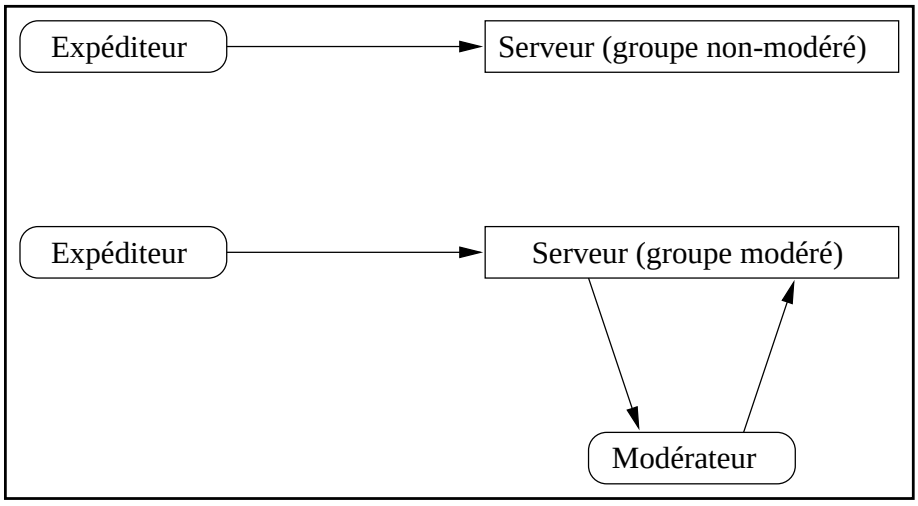

Figure 6.2 *Groupes modérés et non modérés*

La modération présente évidemment des avantages et des inconvénients :

- de par l'élimination des messages inopportuns, la modération augmente grandement le rapport signal sur bruit du groupe de discussion, le rendant en général plus intéressant, notamment lorsqu'il s'agit d'un groupe technique ;
- lorsque le modérateur s'absente pour plusieurs jours, le groupe est bloqué et les requêtes urgentes devront attendre son retour. Il y a également le risque de se brouiller avec le modérateur qui, pour des raisons personnelles, pourra abusivement refuser de poster les nouveaux messages.

6.1.4 Liste des groupes

Comme indiqué page 219, la liste des groupes est bien trop longue pour pouvoir être reprise dans sa totalité. Cependant l'attribution de noms aux groupes respecte certaines conventions permettant de cibler plus précisément la recherche lorsqu'on veut accéder à un groupe traitant d'un sujet particulier.

Le nom d'un groupe est construit à partir d'une structure hiérarchique cohérente ; par exemple, le groupe `fr.comp.os.unix` est un groupe francophone qui a comme sujet l'informatique et plus précisément les systèmes d'exploitation de type Unix. Le nom se décompose ainsi :

`fr:` désigne un groupe francophone ;

`comp:` abréviation du mot anglais « *computing* », approximativement « informatique » ;

`os:` acronyme signifiant « *operating systems* » c'est-à-dire « systèmes d'exploitation » ;

`unix:` Unix.

Le tableau 6.1 donne la signification de quelques-uns des symboles que l'on trouve dans la partie gauche d'un nom de groupe.

Symbole	Signification
`alt`	groupes alternatifs, traitant de tout
`comp`	groupes traitant de sujet informatiques
`de`	groupes germanophones
`fr`	groupes francophones
`gnu`	logiciels sous licence GNU
`it`	groupes de langue italienne
`news`	groupes traitant du système de forum lui-même
`sci`	groupes scientifiques
`soc`	groupes traitant de sujets de société
`uk`	groupes anglophones

Tableau 6.1 *Groupes par catégories*

Il est possible toutefois de trouver une liste des groupes régulièrement remise à jour à l'URL `ftp://garbo.uwasa.fi:/pc/doc-net/newsgrps.zip`. Cette liste ne contient cependant pas tous les groupes, notamment les hiérarchies nationales, comme les groupes dont le nom commence par `fr`.

6.2 Propagation des messages

Les listes de distribution reposent sur un modèle centralisé : le message est envoyé au serveur qui gère la liste, et le serveur le renvoie à tous les abonnés. Si le serveur est hors service, cela signifie qu'aucun des lecteurs de la liste ne pourra recevoir de nouveau message.

Les forums sont basés sur un système totalement différent ; l'ensemble des serveurs Usenet forme un arbre orienté comportant des redondances (voir figure 6.3).

Lorsqu'un utilisateur souhaite poster un message, il le fait auprès de son serveur Usenet le plus proche ; son serveur le propage ensuite à d'autres serveurs, qui eux-mêmes le propagent à d'autres, etc.

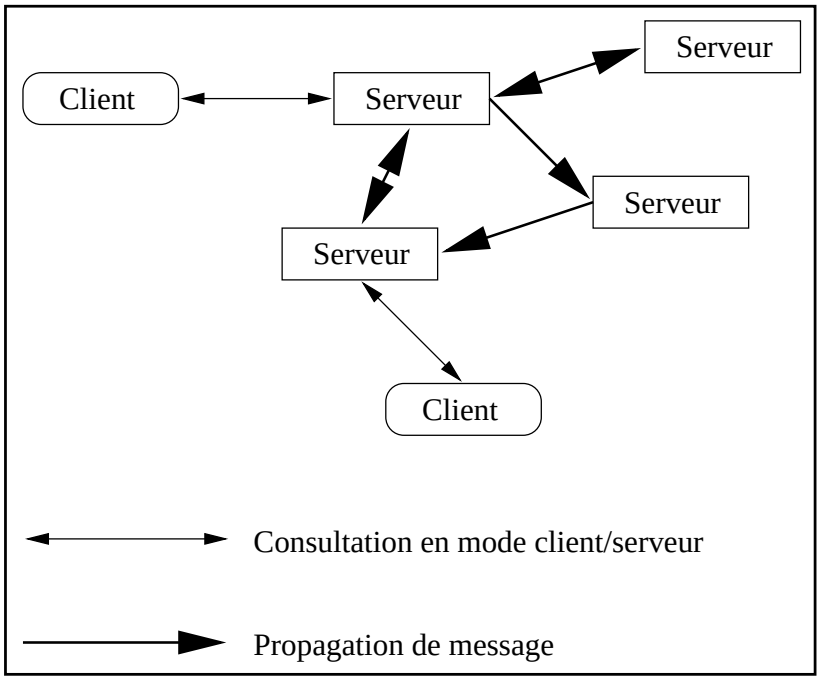

Figure 6.3 *Graphe des serveurs Usenet*

Chaque message possède un identificateur unique, le `Message-Id`. Lorsqu'un serveur A propose à un serveur B un article qui possède un identificateur donné, le serveur B vérifie tout d'abord dans son fichier d'historique qu'on ne lui a jamais présenté ce message auparavant. Si ce message est nouveau, alors le serveur B choisit, en fonction de sa configuration, s'il souhaite accepter ou non le message ; il le marque également comme étant connu dans son fichier d'historique.

C'est ainsi que le réseau Usenet autorise les redondances sans faire de transferts de messages inutiles ; un message peut potentiellement suivre plusieurs chemins, ce qui permet de résister aux mises hors service de certains nœuds du réseau.

La figure 6.4 page suivante montre comment un message est propagé en plusieurs étapes. Le client envoie le message au serveur le plus proche (en général un serveur local), qui le renvoie à d'autres serveurs, qui le renvoient encore à d'autres serveurs (selon la chronologie indiquée par des numéros). Comme on pouvait s'y attendre, les étapes 4 et 4' échouent car les serveurs à qui on propose le message en ont déjà connaissance auparavant.

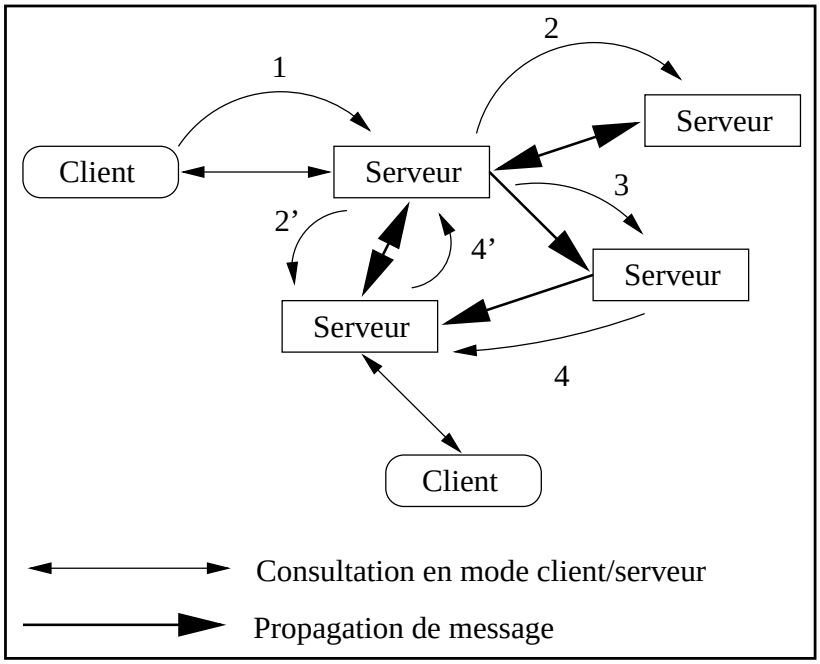

Figure 6.4 *Propagation d'un message*

6.3 Cycle de vie d'un article

Un article posté dans un forum n'est pas disponible *ad vitam aeternam*. Il suit un cycle de vie qui est déterminé par de nombreux paramètres.

6.3.1 Propagation

Tout d'abord, l'article doit être propagé entre les différents serveurs Usenet. Certains serveurs n'acceptent que certains groupes, par exemple les groupes francophones (commençant par `fr`). Si l'on souhaite garder la cohérence de chaque groupe (c'est-à-dire que tous les serveurs transportant un groupe y voient les mêmes articles) il faut s'assurer que le graphe orienté constitué des serveurs transportant ce groupe est connexe.

Toutefois lorsqu'un message est posté en une fois à destination de plusieurs groupes différents, il suffit qu'un serveur transporte un seul de ces groupes pour qu'il accepte le message et le repropage. C'est pourquoi un message posté dans un groupe local à une institution sera quand même propagé à l'extérieur s'il est posté en même temps dans un groupe national par exemple.

Les mécanismes de propagation entre les différents serveurs ne sont pas identiques ; certains serveurs s'échangent des messages en permanence en gardant établie une connexion TCP tandis que d'autres les échangent par paquets en utilisant par exemple UUCP sur une ligne téléphonique. C'est ainsi que certains messages peuvent mettre plusieurs jours pour parvenir

sur tous les sites potentiellement intéressés, et que la réponse à une question peut elle aussi mettre énormément de temps, ce qui nuit parfois à l'interactivité des discussions.

6.3.2 Effacement

Lorsque son auteur le souhaite, un article peut être effacé, notamment lorsque le rédacteur s'aperçoit qu'il a posté des informations erronées ; étant donné la structure répartie des serveurs Usenet, il est impossible d'effacer effectivement cet article de tous les serveurs à la fois.

C'est pourquoi on propage un « message de contrôle », qui est copié de serveur en serveur de la même manière qu'un article. Lorsque le message de contrôle arrive sur un site sur lequel se trouve l'article qu'il doit effacer, il le supprime et l'article devient donc indisponible depuis ce serveur.

Comme indiqué page ci-contre, les temps de propagation sur le réseau Usenet ne sont pas uniformes et les liaisons les plus lentes (connexions par modem) sont généralement des feuilles de l'arbre ; c'est pourquoi, en pratique, un message d'effacement posté peu après le message d'origine a de bonnes chances d'arrêter en cours de route la propagation du message concerné et donc d'économiser de la bande passante au lieu d'en consommer.

6.3.3 Expiration

Le nombre de messages échangés chaque jour est considérable ; il est bien entendu impossible de les conserver tous sans arriver à saturation, quelle que soit la capacité des supports de stockage.

Pour cette raison, Usenet utilise l'**expiration** des messages : chaque message sera, après un temps donné fixé par exemple à deux semaines, effacé de la machine locale.

Cette façon de procéder est avantageuse dans le sens où la place est réservée pour les messages les plus récents ; elle est par contre pénalisante pour l'utilisateur lorsque celui-ci s'absente pendant un temps supérieur à la période d'expiration, ce qui fait qu'à son retour certains messages ne lui seront jamais présentés.

6.4 Configuration d'un lecteur de forums

Il existe de nombreux programmes permettant de lire et écrire dans les forums de discussion. Toutefois, le programme le plus simple à utiliser est assurément `Netscape`, qui permet d'avoir un aperçu rapide de la discussion en cours et de sélectionner à la souris les articles qu'on souhaite lire. La figure 6.5 page suivante montre un exemple de lecture de forums en utilisant `Netscape`.

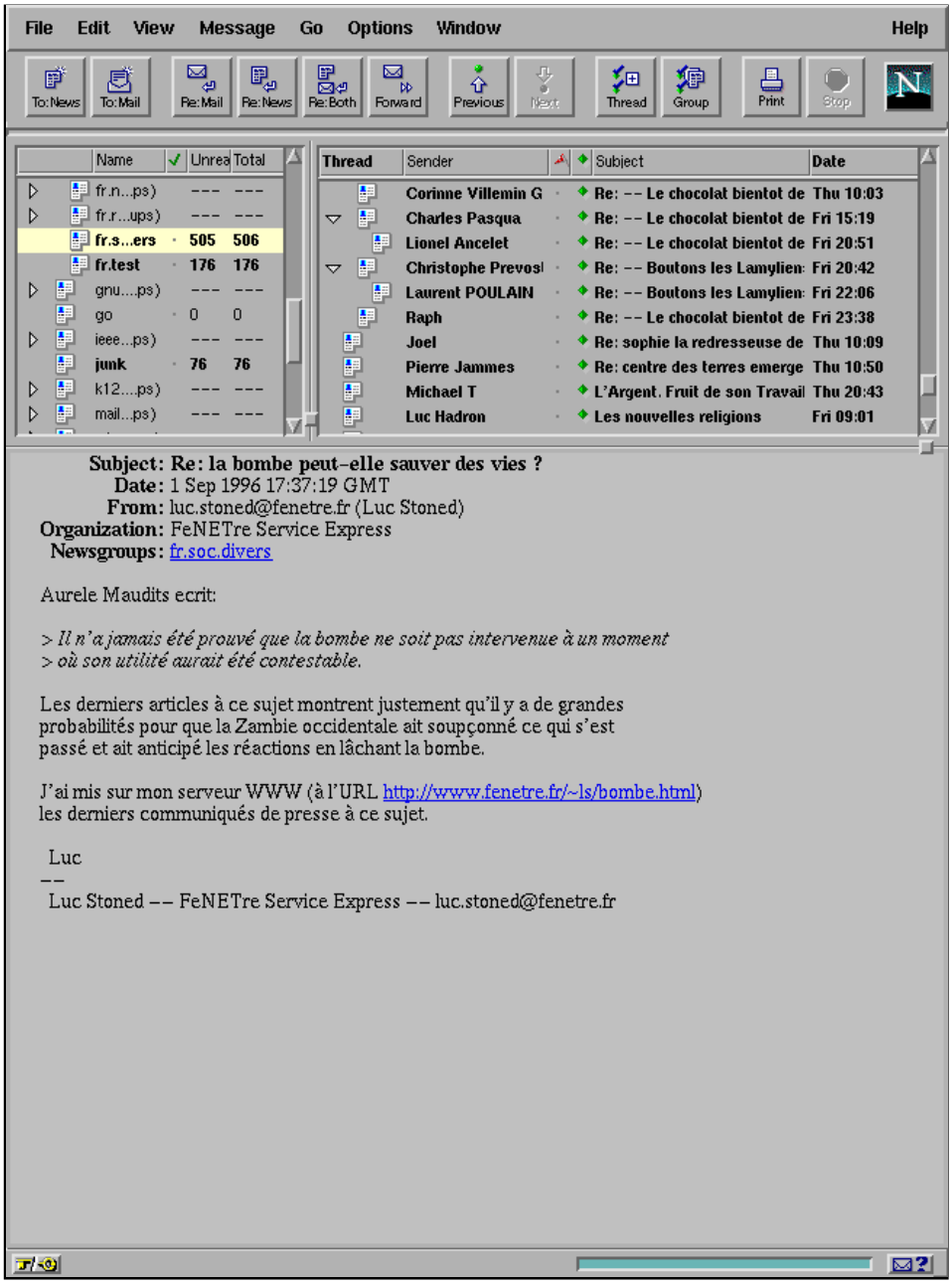

Figure 6.5 *Lecture de forums avec Netscape*

MIME ou 8 bits ?

Il est de très mauvais goût de poster un article à la norme MIME 7 bits dans les forums ; cela vaut en général à l'expéditeur un grand nombre de messages en retour d'utilisateurs insatisfaits d'avoir lu dans leur groupe favori un message cryptique. Pour cela, il faut choisir, dans les préférences de Netscape, de poster en 8 bits afin d'éviter les griefs des autres usagers du réseau.

Afin d'accéder au système de forums, il faut indiquer à `Netscape` l'adresse du serveur ; dans

notre cas, Luc STONED a demandé à accéder à l'URL `news://news.fenetre.fr/` qui est l'adresse du serveur de notre entreprise. À partir de ce moment là, Luc STONED peut s'abonner ou se désabonner de certains groupes en utilisant les menus de `Netscape`.

Comme indiqué page 220, `Netscape` maintient à jour un fichier appelé `.newsrc` dans le répertoire de l'utilisateur. Un extrait de ce fichier est présenté ci-dessous :

```
fr.education.medias: 1-12
comp.lang.ada: 1-41423
comp.lang.python: 1-12561
gnu.announce: 1-560
comp.emulators.announce: 1-122
fr.announce.divers: 1-2578
fr.announce.important: 1-15
fr.announce.newgroups: 1-450
fr.comp.infosystemes: 1-6043
```

Le premier champ de chaque ligne indique le nom du groupe, tandis que les nombres se réfèrent aux articles qui ont déjà été lus. Comme on peut le remarquer, ce ne sont pas les `Message-Id` des articles qui sont enregistrés, mais des nombres qui ne sont en fait qu'une numérotation propre au serveur de forums concerné. Cela signifie que lorsqu'on désire changer de serveur de forums, il faut réinitialiser cette liste au risque de relire une nouvelle fois des articles déjà rencontrés.

6.5 Configuration d'un serveur : INN

Le programme `INN` (InterNet News) a été écrit par Rich SALZ et est maintenu par l'ISC (Internet Software Consortium). La dernière version est la `1.4sec`, qui ajoute à la version `1.4` des programmes plus sécurisés. Il est capable d'échanger des messages en utilisant le protocole NNTP et gère également l'expiration des messages et la gestion des messages de contrôle. Il peut également archiver les messages de certains groupes ou avertir l'administrateur de la présence de certains messages de contrôle spécifiques.

Note importante

De même que pour « surfer » sur le WWW il n'y a aucun besoin d'installer un serveur WWW, il n'est pas nécessaire d'installer un serveur de forums pour lire les articles. Cette partie n'est intéressante que pour les sites qui souhaitent disposer eux-mêmes de leur propre système de forum.

6.5.1 Installation

La compilation d'`INN` n'est absolument pas conviviale ; la configuration n'est pas basée sur `autoconf` à la différence de la plupart des programmes destinés à être installés sur une grande variété de plates-formes. C'est pourquoi il est grandement conseillé de lire les FAQ

(Frequently Asked Questions) d'INN afin de ne pas perdre de temps durant la phase de configuration.

Le programme INN peut être récupéré, ainsi que des utilitaires relatifs à la propagation des forums de discussion, à l'URL ftp://ftp.uu.net/networking/news/nntp/inn/. L'installation débute ainsi :

```
% mkdir inn
% cd inn
% zcat ../inn1.4sec.tar.Z | tar xpBf -
% make Install.ms
cat Install.ms.1 Install.ms.2 >Install.ms
chmod 444 Install.ms
```

Ceci construit la documentation relative à l'installation d'INN. Elle est visualisable en mode texte avec la commande :

```
% nroff -ms Install.ms | more
```

ou, imprimable, lorsque le système dispose de groff par la commande :

```
% groff -ms Install.ms | lpr
```

La configuration se fait en modifiant le fichier config/config.data, qui doit être copié à partir du fichier config/config.dist et en construisant le programme subst :

```
% cd config
% cp config.dist config.data
% chmod u+w config.data
% make c CC=gcc
gcc -o subst subst.c
% vi config.data
```

La première chose à modifier est la section correspondant au compilateur. Lorsque le système dispose de gcc, il est conseillé de l'utiliser.

```
##  C compiler
#### =()<CC                    @<CC>@>()=
CC                      gcc
```

Toute la partie de la configuration concernant l'emplacement des fichiers, les propriétaires et les permissions ont des valeurs par défaut cohérentes. La seule opération à effectuer concernant cette partie est la création d'un compte news ainsi que d'un groupe du même nom ; c'est sous cette identité que tourneront la plupart des programmes qui font partie d'INN. Les répertoires utilisés par INN sont décrits dans le tableau 6.2 page ci-contre.

En ce qui concerne la partie de la configuration spécifique à la machine, il est conseillé de s'inspirer de la documentation fournie avec INN.

Répertoire	Utilisation
`/usr/local/etc`	Programmes tournant en tâche de fond
`/usr/local/news`	Fichiers administratifs de contrôle d'`INN`
`/var/log/news`	Fichiers journaux
`/var/spool/news`	Répertoire contenant les articles eux-mêmes

Tableau 6.2 *Répertoires utilisés par INN*

Certains paramètres permettent de modifier le comportement d'`INN` et des programmes associés ; ils sont décrits dans le tableau 6.3.

Variable	Signification
`CHECK_INCLUDED_TEXT`	Vérifie que les articles contiennent au moins autant de texte original que de texte recopié.
`MAX_ART_SIZE`	Taille maximum en octets d'un article.
`INND_NICE_KIDS`	Il faut définir cette variable si on souhaite que les processus lancés par `INN` ne prennent pas toute la puissance de la machine (cas typique d'une machine non dédiée à Usenet).
`NNRP_LOADLIMIT`	Charge à partir de laquelle `INN` refusera des nouveaux clients (`-1` désactive cette possibilité).
`DEFAULT_CONNECTIONS`	Nombre de connexions NNTP maximum.
`INNWATCH_SPOOLSPACE`	Unités de disque qui doivent rester libres dans `/usr/spool/news`.
`INNWATCH_LIBSPACE`	Unités de disque qui doivent rester libres dans `/usr/local/news`.

Tableau 6.3 *Modification du comportement d'INN*

Il faut ensuite modifier le chemin de certains utilitaires, dont la position dépend de la configuration du site. On peut maintenant compiler `INN` :

```
% make all
[...]
% make install
/bin/sh ./makedirs.sh
+ [ ! -d /usr/local/man/man1 ]
+ [ ! -d /usr/local/man/man3 ]
+ [ ! -d /usr/local/man/man5 ]
+ [ ! -d /usr/local/man/man8 ]
+ [ ! -d /var/spool/news ]
+ [ ! -d /var/spool/news/news.archive ]
+ mkdir /var/spool/news/news.archive
[...]
% cd site
% make install
```

6.5.2 Configuration

Attention

Les fichiers de configuration d'`INN` ne doivent pas être modifiés directement dans le répertoire dans lequel ils sont installés. La modification doit se faire dans le sous-répertoire `site` de la distribution ; les fichiers modifiés seront installés avec la commande `make update`.

`INN` se configure à l'aide de plusieurs fichiers de configuration. Le plus compliqué à gérer est probablement le fichier `neewsfeeds`. Les lignes vides dans les fichiers de configuration sont ignorées et celles commençant par le caractère « # » sont des commentaires.

Le fichier `expire.ctl`

Le fichier `expire.ctl` contrôle l'expiration des articles afin de libérer de la place disque. Il contient des enregistrements de deux formats différents :

Temps de mémorisation d'un article : comme indiqué page 223, un article reste dans l'historique d'un serveur un certain temps, même après son expiration. Ceci est indiqué de la manière suivante dans le fichier de configuration :

```
/remember/:days
```

où `days` contient un nombre de jours (14 signifiera que les `Message-Id` doivent être mémorisés pendant 2 semaines).

Délai d'expiration : des lignes composées de 5 champs au format suivant gèrent l'expiration :

```
pattern:modflag:keep:default:purge
```

Ici, `pattern` désigne des noms de groupes séparés par des virgules et peut contenir des noms génériques (par exemple « `sci.*,fr.soc.*` »), tandis que `modflag` prend une des valeurs indiquées dans le tableau 6.4.

Symbole	Signification
`M`	Groupes modérés
`U`	Groupes non modérés
`A`	Tous les groupes

Tableau 6.4 *Valeurs du champ `modflag`*

Les trois champs suivants, respectivement `keep`, `default` et `purge`, déterminent le délai

d'expiration et le respect ou non de l'en-tête éventuelle du message donnant une date d'expiration : s'il n'y a pas de telle en-tête, alors le message expirera au bout de `default` jours, sinon, il expirera au plus tôt au bout de `keep` jours et au plus tard au bout de `purge` jours, en essayant de satisfaire la valeur indiquée dans l'en-tête.

Un exemple de fichier `expire.ctl` suit :

```
## Garder l'historique pendant 5 jours
/remember/:5
## Respecter au maximum les en-têtes des groupes modérés
*:M:1:30:90
## Les groupes non modérés expirent tous au bout de 3 jours
*:U:3:3:3
```

Le fichier `hosts.nntp`

Ce fichier contient la liste des machines autorisées à envoyer des articles à notre serveur. Chaque ligne de ce fichier contient deux ou trois champs :

1. le nom de la machine distante : ce doit être un nom valide (référencé dans le DNS), par exemple `news2.fenetre.fr` ;
2. le mot de passe à utiliser pour la connexion, qui peut être vide ;
3. la liste des groupes dans lesquels la machine distante peut envoyer des articles (ce champ est facultatif).

Si on reçoit les articles de notre fournisseur `news.fournisseur.fr` et qu'on ne souhaite rapatrier que les groupes de la hiérarchie `fr`, le fichier ressemblera à :

```
## Accepter les groupes fr.* de notre fournisseur
## (sans mot de passe)
news.fournisseur.fr::fr.*
```

Le fichier `inn.conf`

Ce fichier contient les variables de configuration générales de notre système de forums. Tous les champs ont une syntaxe de la forme :

```
nom : valeur
```

où `nom` est choisi dans le tableau 6.5 page suivante.

Exemple de fichier de configuration :

```
## Configuration pour les machines *.fenetre.fr
fromhost:        fenetre.fr
moderatormailer: %s@uunet.uu.net
organization:    FeNETre Service Express
server:          news.fenetre.fr
domain:          fenetre.fr
```

Attribut	Signification
`fromhost`	Nom de la machine locale
`moderatormailer`	Machine à utiliser pour atteindre les modérateurs
`organization`	Nom de la société
`pathhost`	Nom de la machine apparaissant dans l'en-tête `Path`
`server`	Serveur NNTP à utiliser pour poster un article
`domain`	Nom de domaine
`mime-version`	Version du protocole MIME à annoncer
`mime-contenttype`	Type de contenu MIME à annoncer
`mime-encoding`	Codage MIME à annoncer

Tableau 6.5 *Attributs de `inn.conf`*

Le fichier `moderators`

Ce fichier contient la liste des modérateurs associés à chaque groupe. Il faut le demander à son fournisseur d'accès.

Le fichier `newsfeeds`

La génération de ce fichier constitue la partie la plus délicate de la configuration d'`INN`. Étant donné la longueur potentielle des entrées, toute ligne se terminant par le caractère « `\` » se continue sur la ligne suivante, après avoir enlevé les espaces en tête de la ligne de continuation.

Les entrées de ce fichier désignent les machines à qui notre programme va renvoyer les articles, en vue de leur propagation ; chaque entrée est constituée de quatre champs :

1. ce champ contient le nom du site, et, éventuellement, plusieurs motifs séparés par des virgules après une barre oblique (« `/` »). Si le chemin par lequel est passé le message contient déjà un de ces noms, le message ne sera pas propagé. L'entrée `ME` a une signification particulière : elle doit être unique et placée en tête du fichier, et permet de surcharger certaines valeurs de paramètres ;
2. ce champ contient la liste des groupes valides pour lesquels on propage les messages ; s'il est vide, cela signifie tous les groupes. On peut exclure certains groupes en les préfixant par le caractère « `!` ». Les groupes francophones à l'exception des groupes informatiques seront par exemple désignés par le motif « `fr.*,!fr.comp.*` » ;
3. ce champ contient des informations diverses, dont certaines reprises dans le tableau 6.6 page ci-contre. Il est indispensable de consulter la documentation de `newsfeeds` (on peut utiliser la commande « `man newsfeeds` ») pour avoir la liste exhaustive de ces paramètres ;
4. ce champ (appelé par la suite le champ « `param` ») contient des paramètres supplémentaires dépendant du type de canal utilisé.

Symbole	Signification
<*taille*	Taille maximum des articles
Ad	Nécessite une en-tête `Distribution`
Ap	Pas de vérification de l'en-tête `Path`
G*nombre*	Nombre maximum de groupes pour cet article
H*nombre*	Nombre maximum de relais pour cet article
Nm	N'envoie que les articles dans les groupes modérés
Nu	N'envoie que les articles dans les groupes non modérés
T*lettre*	Type de canal à utiliser

Tableau 6.6 *Informations de propagation*

Les différents types de canaux sont les suivants :

Le type vide (log feed). Ce type de canal ne fait rien d'autre que d'écrire une ligne dans le fichier journal.

Le type fichier (file feed). Ce type de canal fait que `INN` se contente d'écrire dans un fichier dont le nom est donné en paramètre dans le champ `param` une ligne par article à poster à l'extérieur. Si aucun nom de fichier n'est donné, alors le nom choisi est par défaut `/usr/spool/news/out.going/`*nom du site*.

Le type programme (program feed). À chaque fois qu'un article est disponible, le programme dont le nom est indiqué dans le champ `param` est lancé ; la succession de symboles « `%s` » sera remplacée par le chemin menant à l'article relativement à `/usr/spool/news`.

Le type canal (channel feed). Le champ `param` désigne ici un programme qui sera lancé une fois pour toutes et auquel on enverra sur l'entrée standard une ligne à chaque fois qu'un nouvel article sera disponible. Si le programme se termine, il sera relancé automatiquement.

Un exemple de fichier `newsfeeds` est présenté ci-dessous :

```
## Ne jamais renvoyer ce qu'il y a dans les groupes locaux
ME:*,!fenetre.*\
    ::
##
## Envoyer tous les articles à notre fournisseur d'accès
news.fournisseur.fr/fournisseur\
    :*\
    :Tf,Wnm:
##
## Archiver tout ce qui passe dans fr.soc.divers en utilisant
## le programme /usr/bin/meta-archive
archive\
    :!*,fr.soc.divers\
    :Tp,Nm:/usr/bin/meta-archive %s
##
## Archiver tout ce qui passe dans soc.culture.french en utilisant
## le programme /usr/bin/archive-and-forward
archive\
    :!*,soc.culture.french\
    :Tp,Nm:/usr/bin/archive-and-forward %s
```

Le fichier `nnrp.access`

Ce fichier contrôle les accès en mode lecture au serveur de forums. Les lignes sont composées de cinq champs :

1. nom de la machine distante ; ce champ peut contenir des caractères génériques, comme « `*.fenetre.fr` » ;
2. permissions ; `R` signifie que le client peut lire les articles, `P` qu'il peut en envoyer. La présence des deux lettres indique que le client peut à la fois lire et envoyer des articles à travers ce serveur ;
3. nom de l'utilisateur lorsque l'authentification est utilisée ;
4. mot de passe de l'utilisateur lorsque l'authentification est utilisée ;
5. liste des groupes auxquels l'utilisateur peut accéder.

Si on souhaite par exemple que nos collègues de la société « Toiture 2000 » (domaine `toiture.fr`) puissent lire les forums depuis notre serveur, il suffira d'utiliser ce qui suit comme `nnrp.access` :

```
## Accès total depuis notre site
*.fenetre.fr:RP:::*
## Accès total depuis notre filiale italienne
*.fenetre.it:RP:::*
## Lecture uniquement pour nos voisins, sauf
## pour nos groupes locaux dont l'accès est
## interdit
*.toiture.fr:R:::*,!fenetre.*
```

Le fichier `passwd.nntp`

Ce fichier contient des lignes de quatre champs contenant les noms et mots de passe à utiliser lorsque notre serveur est contacté par un client :

1. nom du client ;
2. nom à utiliser lorsqu'on se présente ;
3. mot de passe à utiliser ;
4. type d'authentification (optionnel).

6.6 Créer un nouveau groupe francophone

La procédure de création d'un groupe varie selon les hiérarchies. Nous allons détailler ici les étapes à suivre pour créer un groupe francophone dans la hiérarchie `fr`. La procédure de création est postée mensuellement dans le groupe `fr.usenet.groups` par Christophe WOLFHUGEL.

6.6.1 Discussion préalable

Avant toute chose, un appel à discussion[1] doit être posté dans le groupe de discussion nommé `fr.announce.newgroups` ainsi que dans tous les groupes dont le sujet est proche de celui du groupe qu'on souhaite créer. Il est bien entendu possible de rendre l'article plus largement accessible, en le plaçant par exemple sur une page WWW ou en l'envoyant sur une liste de diffusion traitant d'un sujet connexe. Il faut également définir l'en-tête `Followup` de telle manière que la discussion qui s'en suivra se déroule dans le forum `fr.usenet.groups`.

L'article d'origine doit contenir la description détaillée du groupe ainsi que son type (modéré ou non modéré) ; s'il est modéré, le nom du modérateur devra également être précisé. Il est conseillé de choisir un nom de groupe s'intégrant dans la hiérarchie existante afin de garder une structure de noms cohérente.

Si un consensus en faveur de la création de ce groupe se dégage après un délai d'au plus 30 jours, alors la personne proposant la création du groupe doit poster un appel à voter[2] dans les mêmes groupes que l'appel à discussion.

6.6.2 Organisation du vote

Cet appel à voter doit contenir de manière claire les instructions à suivre pour voter en faveur ou contre la création du groupe. La période de vote doit durer entre 21 et 31 jours inclus, même si les résultats intermédiaires laissent deviner qu'il n'y aura pas de surprise quant au résultat final.

Durant la période de vote, il est possible de poster un récapitulatif des votes reçus **mais sans en donner le contenu**, c'est-à-dire que le nom des personnes ayant voté peut être publié (cela permet aux différents participants de vérifier que leur vote a bien été pris en compte) mais que les choix individuels ou le score en faveur ou contre la création du groupe ne doit pas être divulgué.

6.6.3 Création véritable du groupe

Si, à l'issue de la période de vote, il y a au moins deux tiers des votes en faveur de la création du groupe et au moins 30 votes de plus en faveur de la création que contre, alors la création du groupe est considérée comme acquise. Le nom de tous les votants ainsi que leur vote doit être posté dans la liste de groupes utilisée ci-avant, afin que chacun puisse vérifier que son vote a bien été pris en compte correctement.

Après un délai d'une semaine, s'il n'y a pas d'objection susceptible d'invalider le vote, le modérateur de la hiérarchie `fr` envoie un message de création de groupe ; dès que les différents

1. En anglais, *Request For Discussion* ou RFD
2. En anglais, *Call For Vote* ou CFV

serveurs de forums ont pris ce message en compte, le groupe peut être utilisé et sera propagé normalement. Si le groupe doit être modéré, le modérateur doit envoyer ses coordonnées électroniques à l'adresse `fr-announce-newgroups@grasp.insa-lyon.fr` afin de configurer correctement les paramètres du groupe.

Lorsque la création d'un groupe a été rejetée par le biais d'un vote, aucune tentative de création de ce groupe ne peut avoir lieu avant un délai de 6 mois suivant la clôture du vote.

TROISIÈME PARTIE

Services multimédias

Le World Wide Web et les services multicast mettent le multimédia à la portée des Internautes : balade sur les serveurs Web, participation à des vidéo-conférences et travail coopératif sont maintenant des activités courantes sur l'Internet.

7

Échange de fichiers

La plupart des services multimédias de l'Internet sont fondés sur des formats d'échange bien définis, dont la vocation est, soit de permettre la transmission de données binaires sur des vecteurs de communication qui ne sont pas conçus pour cet usage, soit d'encapsuler les informations à transmettre dans des messages qui recensent leur type et leur provenance.

Différentes techniques de codage sont utilisées, selon qu'il s'agisse de transférer des fichiers par *email*, par le *news* ou encore par le Web. De plus, il est souvent préférable d'archiver, puis de compresser, des fichiers ou des groupes de fichiers de taille conséquente.

Ce chapitre présente les procédés les plus communs utilisés sous Unix. Il détaille également le standard MIME, dont l'importance est certaine, puisqu'il sous-tend l'ensemble des communications du protocole HTTP utilisé sur le Web.

7.1 Encodage des fichiers

Une partie des outils de communication traditionnels de l'Internet est fondée sur la transmission ou la réception d'informations purement textuelles, à partir du jeu de caractères ASCII le plus simple : lettres (sans accents), chiffres et signes de ponctuation ou séparateurs. Ces caractères sont codés sur 7 bits, ils représentent la partie basse du jeu de caractère ASCII étendu aujourd'hui utilisé sur les ordinateurs personnels. Les caractères qui précédent l'espace ainsi que certains séparateurs sont considérés comme des codes de contrôle ou des commandes et revêtent généralement une signification particulière.

Pour transmettre des fichiers quelconques (images, bases de données...) il est bien sûr indispensable de se libérer de cette contrainte du mode texte. Dans le cas de protocoles de transfert

adaptés, tel FTP, la communication peut s'établir en mode binaire, permettant ainsi le passage de tous les codes possibles sur 8 bits. En revanche, d'autres protocoles prévus à l'origine pour ne transférer que des messages en clair ne peuvent reproduire directement ces codes sans risquer de déformer le contenu du message, voire l'usage qui en est fait par les outils chargés de les traiter (ou de les afficher). C'est le cas des protocoles SMTP pour le courrier électronique et NNTP pour les news.

Pour faire transiter un fichier texte *étendu* ou un fichier binaire par l'intermédiaire de ces protocoles, on fait appel à une méthode d'encodage qui ramène tous les codes dans un *intervalle* autorisé (par exemple le jeu de caractères standard de `A` à `Z`, de `a` à `z` et de `0` à `9`). Après transformation, le fichier obtenu est de taille plus importante, puisqu'un caractère *illégal* en mode texte doit être par exemple codé sur deux caractères *légaux* successifs.

Même dans le cas d'un fichier texte, un tel codage peut s'avérer nécessaire, afin de permettre une interprétation correcte des caractères accentués. Le standard MIME, tout comme le standard HTML (le codage utilisé pour le Web), définissent des séquences de caractères spéciaux, dites séquences d'échappement, pour représenter les accents sous forme d'une série de caractères habituels.

7.1.1 MIME (Multipurpose Internet Mail Extensions)

Le standard MIME définit un ensemble de règles pour la mise en forme, la transmission, et l'analyse de messages pouvant contenir n'importe quel types de données.

En particulier, il offre la possibilité d'intégrer plusieurs documents dans un même message, de transmettre du son, des images, de la vidéo, en évitant les pertes de données dues à la transmission de caractères illégaux, et il définit une syntaxe pour représenter un texte rédigé dans plusieurs polices de caractères différentes en conservant les polices en question.

Pour ce faire, un message MIME encapsule les documents à transmettre dans des éléments (*entities*) composés chacun d'un en-tête (*header*) et du document (*body*) concerné. L'en-tête définit le type de média du message, et toute autre information servant à extraire le document et à l'interpréter correctement. Cette encapsulation est récursive, c'est-à-dire qu'il est possible d'encapsuler un message MIME dans un autre message MIME, sans limite théorique sur le nombre de niveaux d'encapsulation.

Il s'agit d'un standard totalement ouvert, librement extensible : il est tout à fait possible d'y ajouter un nouveau type de document – même si un tel type ne peut être officiellement reconnu comme appartenant au standard qu'après l'accord de Iana.

Champ Content-type

Le type d'un document est définit par le champ `Content-type` de l'en-tête qui l'accompagne. Il est obligatoirement composé de deux parties, respectivement le type principal et le

sous-type, et peut être suivi de paramètres facultatifs qui précisent tel ou tel aspect du type en question :

```
Content-type: image/gif
```

Le type représente en quelque sorte la *famille* à laquelle appartient le document. Il peut s'agir d'une image (*image*), d'un texte (*text*), d'un son (*audio*), etc.

Le sous-type indique le format du document. Il permet ainsi de distinguer les différents formats de fichiers images ou vidéo, de la même manière que par l'extension du nom de fichier.

L'ensemble type/sous-type permet de savoir quelle utilisation doit être réservée au document. Ainsi, un message contenant du texte et une image pourra être affiché sous forme d'une page unique si l'outil de visualisation le permet. On comprend que le protocole HTTP utilisé sur le Web se fonde sur le standard MIME, puisqu'une des caractéristiques principales des navigateurs HTML est de pouvoir afficher simultanément du texte, des images, voire des animations ou des séquences vidéos.

La section 7.6.1 page 250 décrit la configuration des types MIME dans Netscape Mail.

Sept types principaux sont définis dans le RFC 1341, document de référence concernant le standard MIME. La liste ci-dessous ne cite que quelques sous-types à titre d'exemple, en particulier les sous-types fondamentaux définis dans le RFC en question.

`multipart`

Un document composite incluant plusieurs parties de formats différents. C'est le type généralement utilisé pour *attacher* un fichier à un *email* (par exemple, joindre une feuille de calcul à un message texte : dans ce cas, le message `multipart` inclura un texte de type `text/plain` et un document Excel dont le type pourrait être `application/excel`).

`mixed`

Un document contenant plusieurs parties de type différent, à afficher de manière séquentielle.

`alternative`

Un document contenant plusieurs versions des mêmes données (par exemple, une version Mac, une version PC...).

`parallel`

Un document composite dont les parties devraient être affichées simultanément.

`digest`

Un document contenant plusieurs parties de type message. La différence entre un message `multipart/mixed` et un message `multipart/digest` est parfois assez subtile puisque les différentes parties d'un message `multipart/mixed` sont

généralement dotées d'un champ `Content-type`. Essentiellement, elle réside dans le fait qu'un message au sens de l'Internet ne peut pas toujours se concevoir comme un document défini et localisé (un fichier). Par ailleurs, un message ne contient pas nécessairement des données, mais peut plutôt être une référence à un emplacement où trouver ces données. La notion de message (au sens ARPA) est définie dans le RFC 822.

`message`

Un message encapsulé dans le message courant. Comme nous l'avons déjà dit, la notion d'encapsulation est récursive.

`partial`

Ce sous-type est destiné aux messages qui ont été découpés en plusieurs parties, afin par exemple de faciliter l'envoi par email d'un message trop long pour passer en une seule fois.

`external-body`

Ce type de message désigne une référence à un document situé quelque part sur l'Internet. Il permet en particulier d'éviter de transférer un gros volume de données par email alors qu'il serait préférable de les récupérer par FTP, tout en précisant quand même dans un courrier électronique où se trouvent ces données.

`text`

Un document texte, de n'importe quel format. Il peut s'agir d'un simple texte ASCII, d'un document `rtf`, d'un document *WinWord*, etc.

`plain`

Un document texte lisible tel quel, sans qu'il soit nécessaire de faire appel à une application spécifique, traitement de texte ou autre. Seul le jeu de caractères peut varier d'un document à l'autre, il est indiqué en paramètre.

Exemple :

```
Content-type: text/plain; charset=iso-8859-1
```

Le document contient du texte pur, et le jeu de caractères utilisé est l'ISO latin-1 ou ISO 8859-1, le plus commun sur l'Internet.

`html`

Un document destiné au Web (le format HTML est décrit en détail au chapitre 10 page 319).

`image`

Un fichier image, c'est-à-dire tout document dont l'affichage ou l'impression nécessite un périphérique graphique.

`gif`, `jpeg`, `tiff`...

Des fichiers images dans l'un de ces formats.

`audio`

Tout document (sonore) qui nécessite un périphérique audio et un organe de restitution (haut-parleur, écouteurs, téléphone...).

`basic`

Les sons échantillonnés tels quels ou les fichiers Unix (`.au`, `.snd`...)

`video`

Toute image animée éventuellement sonorisée. Ce type n'inclut cependant pas les animations réalisées par programme (avec *Java* ou *Shockwave*) qui possèdent alors le type `application`, ni le GIF animé qui rentre dans la catégorie `image`.

`mpeg`

Les fichiers au format MPEG, sorte de *jpeg* animé.

`application`

N'importe quel type de données qui ne rentre pas dans les types précédents. D'une manière générale, il ne devrait pas être nécessaire de créer un nouveau type, puisque le type application peut recevoir tous les sous-types spécifiques. Dans la pratique, il est cependant permis de créer un nouveau type : par exemple, on a défini `x-world` pour la catégorie des fichiers décrivant un environnement de réalité virtuelle (`x-vrml` en est un sous-type).

`octet-stream`

Ce sous-type particulier désigne un fichier binaire pour lequel aucun traitement particulier n'est souhaité. Le comportement par défaut est alors d'enregistrer le fichier sur disque, éventuellement en demandant un nom de fichier à l'utilisateur.

`postscript`

Pour visualiser un fichier PostScript, il faudra généralement une application extérieure, par exemple `ghostscript`.

On pourra également filtrer le document à travers *Acrobat Distiller* qui produira un fichier PDF (à consulter avec *Acrobat Reader*). Cependant la plupart du temps ce type de fichier sert à l'impression ; selon le fonctionnement de l'outil de mise en page il pourra être nécessaire d'écrire le fichier sur disque avant de l'envoyer à l'imprimante.

Pour créer son propre type, il est recommandé de définir un nouveau sous-type d'un des 7 types principaux. Le nom de ce sous-type doit être précédé du signe `x-` qui indique que le type n'est pas officiel, c'est-à-dire qu'il n'a pas été reconnu par IANA comme faisant partie du standard.

Netscape a par exemple défini le type `multipart/x-mixed-replace` pour certaines applications du Web. Ce type indique que le message contient plusieurs parties qui doivent être affichées au fur et à mesure de leur réception, chacune devant remplacer la précédente. La technique dite *server-push*, qui permet d'insérer très simplement des images animées au sein d'une page Web, utilise ce type de message.

7.1.2 Échange de fichiers codés avec MIME

Champ Content-Transfer-Encoding

Ce champ MIME peut être utilisé dans les en-têtes de messages afin d'indiquer quel procédé d'encodage a été appliqué aux données contenues dans le document. Le standard MIME n'impose pas de procédé particulier, mais tente de limiter le nombre de procédés et propose en particulier quelques solutions qui sont maintenant d'usage courant.

`7bit`, `8bit` et `binary` désignent des documents *non encodés*, ils servent en tant que simple indication des codes qu'on peut s'attendre à trouver dans la suite du message.

Un document `7bit` ne contiendra que du texte US-ASCII (sans accents!), c'est-à-dire le jeu de caractères standard des terminaux alphanumériques, et que la longueur des lignes de texte est limitée (ce qui peut être important si les applications chargées d'acheminer le message supposent une telle limitation : c'est le cas pour les transferts en SMTP).

Un document `8bit` pourra contenir des caractères dont le code est supérieur à 127, mais la longueur des lignes sera toujours limitée, tandis qu'un document `binary` risque de contenir des lignes trop longues pour certaines applications chargées de transférer le message. Ces valeurs ne devraient actuellement pas être utilisées avec les *email* ou les *news* puisque il n'est pas prévu que les messages échangés par ces voies puissent contenir des données sur huit bits. Les clients tels que *Netscape Mail* proposent une option à définir pour autoriser ou interdire l'utilisation du format `8bit`.

L'encodage `base64` fournit un résultat comparable à celui obtenu avec `uuencode` (voir la section 7.3 page 246). Les caractères sont codés pour être ramenés dans la plage des caractères ASCII affichables, et les lignes de texte obtenues sont limitées à 60 caractères. Pour 3 bits en entrée, on obtient 4 bits en sortie. Ce qui, si on ajoute l'information de contrôle et les

sauts de ligne, représente environ 35% d'inflation en taille. Tous les caractères sont encodés systématiquement, y compris le texte pur.

Exemple de fichier compressé avec `gzip`, puis codé en `base64` par *Netscape Mail*

```
Content-Type: application/x-gzip
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename="fichier.gz"

H4sICJvZdDEAAzA1YXByMTk5NgDs/WuP5MiRJox+L+D9D4MX6P6izEjSeW+goaNZqbU7uzuj
o9HszuDFQYIZwcxkVUQwmmRkVunD/vbjTkZkRrqZu5u5M0o6ODOj7rp0lT30m7m5XR7bfuv2
sYhWm7FfPfb/cCv///+JsrvfHfq7uKryn6Jo+p/4h99EcRT9f/7h//7jH/7yD3cP9dOxGcfm
bt3tdt3+rt2Pd9t2GFdP7eM//Ne//OVPd/Eq+r//QUTRP4gkij4dDofbqrwVcbnaN8d2u/22
Wm+PD7fy7zX9vhlt0LkOvd809fH8Y4VAxqKKyzfMJOVjFh8xd23f3B266W9ieGmVhgwxTj7C
PXb79djKae2b9XPTy39Wz+Nuq6GmUSpCBhln1olN0YmV//ep6+W3rV67frtRCPIfKsTwbaw3
7f7u0K6Hu0PdtwMGEuUiPg8sFtnSAxMIZpGX1adxKG4PiTwKbb09Hlbta/3NhpJbxvbU7Ju+
3mKjS5KyDNoslWGzyL92/9xsD+jw4uRtRquEjSmij5hgJ2bZJxFlqzjPV3EiViJJjKLij6IO

[...]
```

Figure 7.1 *Exemple de fichier compressé avec gzip puis codé en base64*

Le champ `Content-type` précise que le fichier doit être envoyé à la commande `gzip` (à condition que l'application qui doit lire ce message ait été correctement configurée dans les associations extensions de fichiers vers types MIME).

Le champ `Content-transfer-encoding` indique quant à lui qu'il faut décoder le message `base64` avant toute utilisation.

Le champ `Content-disposition` a été ajouté par *Netscape Mail*, il indique en particulier que le contenu du fichier attaché au message a été inséré directement dans l'*email* (par opposition à un lien ou à un message encapsulé) et que son nom initial était `fichier.gz`.

Il est possible de définir de nouveaux procédés d'encodage et de les préciser dans le champ `Content-Transfer-Encoding`, à condition de les faire précéder du signe `x-`. Tous les autres noms sont réservés par IANA et sont donc interdits.

7.1.3 Codage des messages avant leur envoi

Le protocole SMTP ne prévoit pas que des messages *email* contiennent autre chose que des caractères ASCII. Dès qu'on souhaite transmettre à travers le réseau un document contenant, soit des accents, soit des données binaires, et si l'on n'est pas sûr que les différents noeuds traversés par le message ne risquent pas de déformer le message en clair, il faut passer par un système d'encodage.

Les clients récents, tels *Netscape Mail* et *Eudora Pro*, encodent automatiquement les documents avant de les envoyer. Il n'est donc plus nécessaire de se préoccuper de ces problèmes, sauf pour s'assurer que le destinataire du message sera en mesure de le décoder.

7.2 Codage des accents selon le standard MIME

7.2.1 Le format Quoted-Printable

Dans le standard MIME, le champ `Content-Transfer-Encoding` peut prendre une valeur particulière destinée aux documents composés pour une majeure partie de texte normal, enrichi de quelques caractères spéciaux ou accentués.

C'est cet encodage particulier, `Quoted-Printable`, qui sera utilisé pour transmettre un texte. Il permet d'éviter l'utilisation pour un document texte de l'encodage `base64`, qui augmente singulièrement la taille du document, quel que soit son type. En revanche, s'il peut également être appliqué à des documents binaires, on lui préférera dans ce cas le format `base64`.

Avec cette technique, les caractères spéciaux sont remplacés par la séquence d'échappement `=nn` où *nn* est le code du caractère en notation hexadécimale. De plus, la longueur maximale d'une ligne est de 76 caractères : si une ligne s'avère être plus longue, elle doit être coupée en plusieurs lignes se terminant par le signe `=`. Les sauts de ligne sont représentés par la séquence CR+LF (retour chariot + saut de ligne), standard sous DOS/Windows mais qui apparaît sous Unix comme un double saut de ligne.

Ainsi le texte suivant :

> Plus loin dans la vallée se trouve une cahute. Pas de vitres aux fenêtres, pas de serrure aux portes, chacun peut y entrer, s'y plaire, et y dormir.

sera transformé en :

> Plus loin dans la vall=E9e se trouve une cahute. Pas de vitres aux =
> fen=EAtres, pas de serrure aux portes, chacun peut y entrer, s'y plaire, =
> et y dormir.

Notons que bien évidemment le signe `=` doit lui-même être encodé (en `=3D`).

7.3 Échange de fichiers codés avec uuencode/uudecode

La commande `uuencode` disponible sous Unix permet d'encoder un fichier selon des règles apparentées à celles utilisées en codage `base64`. Tous les caractères du fichier résultant sont affichables sur un terminal classique, et les lignes ne dépassent pas 60 caractères (61 avec le premier caractère de contrôle).

```
begin 644 websites
M2F%V82‘F(%9234P*"D-U<G)E;G0@5E)-3"‘R+C‘@05!)"G)E86QI=’DN<V=I
M+F-O;2]E;7!L;WEE97,O9V%V:6XO=G)M;"]!<&DN:’1M;‘I#=7)R96YT(%92
M34P@,BXP(’!R;W!O<V%L"G)E86QI=’DN<V=I+F-O;2]E;7!L;WEE97,O9V%V
M:6XO=G)M;"]"96AA=FEO<G,N:’1M;‘I!8W1I=F564DU,("A-:6-R;W-O9G0I
M(’!R;W!O<V%L"G=W=RYM:6-R;W-O9G0N8V]M+VEN=&1E=B]T96-H+FAT;0I3
M1TD@<F5S<&]N<V4@=&\@06-T:79E5E)-3‘IR96%L:71Y+G-G:2YC;VTO96UP
K;&]Y965S+V=A=FEN+W9R;6PO06-T:79E5E)-3%)E<W!O;G-E+FAT;6P*"FUP
‘
end
```

Figure 7.2 *Exemple de fichier codé avec uuencode*

Comme on peut le voir sur l’exemple 7.2, un fichier codé avec `uuencode` commence par un en-tête de la forme `begin nnn nom.de.fichier` et se termine par une ligne `end`. Le nombre *nnn* est exprimé en octal : il s’agit tout simplement des permissions associées au fichier encodé.

La commande prend un ou deux paramètres, respectivement le fichier à encoder et le nom de fichier à inscrire dans l’en-tête. Le nom du fichier à encoder est facultatif, s’il n’est pas précisé c’est l’entrée standard qui sera utilisée. Quant au nom de fichier, sa fonction est d’indiquer à la commande `uudecode`, l’inverse de `uuencode`, sous quel nom le fichier doit être extrait.

La commande `uudecode` lit le fichier dont le nom est passé en paramètre et extrait les informations décodées dans le fichier dont le nom est précisé dans l’en-tête.

```
gide: /encode$ cat texte
Plus loin dans la vallée se trouve une cahute. Pas de vitres aux fenêtres, pas de
serrure aux portes, chacun peut y entrer, s’y plaire, et y dormir.
gide: /encode$ uuencode texte vallee.txt > vallee.uu
gide: /encode$ cat vallee.uu

begin 644 vallee.txt
M4&QU<R!L;VEN(&1A;G,@;&$@=F%L;.EE(’-E(’1R;W5V92!U;F4@8V%H=71E
M+B!087,@9&4@=FET<F5S(&%U>"!F96[J=’)E<RP@<&%S(&1E(’-E<G)U<F4@
M875X(’!O<G1E<RP@8VAA8W5N(’!E=70@>2!E;G1R97(L(’,G>2!P;&%I<F4L
.(&5T(’D@9&]R;6ER+@IA

end

gide: /encode$ uudecode vallee.uu
gide: /encode$ cat vallee.txt
Plus loin dans la vallée se trouve une cahute. Pas de vitres aux fenêtres, pas de
serrure aux portes, chacun peut y entrer, s’y plaire, et y dormir.
gide: /encode$ tar cf - projet_web | gzip -9 | uuencode projet_web.tar.gz | mail
rousseau@fenetre.fr
```

Figure 7.3 *Utilisation des commandes uuencode et uudecode*

7.4 Utilisation du jeu de caractères ISO-latin-1

Avec la généralisation des caractères accentués, une norme ISO s'est imposée : le jeu de caractères `iso-latin-1` (ou `iso-8859-1` ou ANSI). C'est ce jeu qui est aujourd'hui largement exploité dans les *emails* et sur le Web, sans qu'il soit nécessaire d'utiliser un encodage spécial.

Il est supporté sur les systèmes de type Unix grâce à de nombreux outils permettant d'adapter les différentes applications d'affichage, et c'est le jeu de caractères standard sous Windows (ceux de MS-DOS, OS/2 et MacOS sont en revanche sensiblement différents).

7.5 Récupération de données provenant de systèmes dont le jeu de caractères est incompatible

Il est courant de rencontrer quelques problèmes de lisibilité lors de la réception d'un message provenant d'une machine tournant sous un système différent : c'est le cas sous Windows à la lecture d'un *email* écrit sur Macintosh ou OS/2.

Sur la figure 7.4, les caractères anormaux ont été remplacés par des traits de soulignement.

Version codée MIME sous OS/2

```
Je vous confirme l'enregistrement de votre =
soci=82t=82 =85 notre s=82minaire =
technique=20
du 15 juillet.
```

Affichage sous Windows

```
Je vous confirme l'enregistrement de votre soci_t_ _ notre
s_minaire technique du 15 juillet.
```

Figure 7.4 *Incompatibilités entre les jeux de caractères*

La solution à ce type de problème est soit de demander à l'expéditeur d'utiliser le jeu de caractères `iso-latin-1`, généralement disponible dans la configuration des outils d'édition d'*emails*, soit de mettre en place un utilitaire de 'traduction'.

Certaines applications permettent de lire des fichiers texte provenant d'autres systèmes en convertissant les caractères accentués correctement, mais ce type de manipulation peut s'avérer fastidieux. Dans tous les cas, il reste indispensable de savoir sous quel système a été rédigé le document...

La liste des principaux jeux de caractères qu'on peut être amené à rencontrer se trouve dans le RFC 1345.

ISO-latin-1		Macintosh	
□	128	-	173
□	129	°	176
‚	130	â	226
ƒ	131	Ä	196
„	132	ã	227
…	133	É	201
†	134		160
‡	135	à	224
ˆ	136	ö	246
‰	137	ä	228
Š	138	²	178
‹	139	Ü	220
Œ	140	Î	206
□	141	³	179
□	142	¶	182
□	143	·	183
□	144	¸	184
‘	145	Ô	212
’	146	Õ	213
“	147	Ò	210
”	148	Ó	211
•	149	¥	165
–	150	Ð	208
—	151	Ñ	209
˜	152	÷	247
™	153	ª	170
š	154	¹	185
›	155	Ý	221
œ	156	Ï	207
□	157	º	186
□	158	½	189
Ÿ	159	Ù	217

ISO-latin-1		Macintosh	
	160	Ê	202
¡	161	Á	193
¢	162	¢	162
£	163	£	163
¤	164	Û	219
¥	165	´	180
¦	166	Ã	195
§	167	¤	164
¨	168	¬	172
©	169	©	169
ª	170	»	187
«	171	Ç	199
¬	172	Â	194
-	173	Å	197
®	174	¨	168
¯	175	Æ	198
°	176	¡	161
±	177	±	177
²	178	×	215
³	179	Ú	218
´	180	«	171
µ	181	µ	181
¶	182	¦	166
·	183	á	225
¸	184	ü	252
¹	185	Þ	222
º	186	¼	188
»	187	È	200
¼	188	ß	223
½	189	ð	240
¾	190	õ	245
¿	191	À	192

ISO-latin-1		Macintosh	
À	192	Ë	203
Á	193	ç	231
Â	194	å	229
Ã	195	Ì	204
Ä	196	□	128
Å	197	□	129
Æ	198	®	174
Ç	199	‚	130
È	200	é	233
É	201	ƒ	131
Ê	202	æ	230
Ë	203	è	232
Ì	204	í	237
Í	205	ê	234
Î	206	ë	235
Ï	207	ì	236
Ð	208	ø	248
Ñ	209	„	132
Ò	210	ñ	241
Ó	211	î	238
Ô	212	ï	239
Õ	213	Í	205
Ö	214	…	133
×	215	ù	249
Ø	216	¯	175
Ù	217	ô	244
Ú	218	ò	242
Û	219	ó	243
Ü	220	†	134
Ý	221	ú	250
Þ	222	û	251
ß	223	§	167

ISO-latin-1		Macintosh	
à	224	ˆ	136
á	225	‡	135
â	226	‰	137
ã	227	‹	139
ä	228	Š	138
å	229	Œ	140
æ	230	¾	190
ç	231	□	141
è	232	□	143
é	233	□	142
ê	234	□	144
ë	235	‘	145
ì	236	”	147
í	237	’	146
î	238	”	148
ï	239	•	149
ð	240	ý	253
ñ	241	–	150
ò	242	˜	152
ó	243	—	151
ô	244	™	153
õ	245	›	155
ö	246	š	154
÷	247	Ö	214
ø	248	¿	191
ù	249	□	157
ú	250	œ	156
û	251	□	158
ü	252	Ÿ	159
ý	253	þ	254
þ	254	ÿ	255
ÿ	255	Ø	216

Tableau 7.1 *Correspondance entre les jeux de caractères iso-latin-1 et mac*

Il existe un utilitaire GNU pour transformer des fichiers en passant d'un jeu à l'autre. Son nom est `recode` (il est accessible par FTP sur les archives GNU ou les sites miroirs, par exemple `ftp.ibp.fr/gnu/`). Il reconnaît la plupart des jeux de caractères décrits dans le RFC 1345, soit près de 150 jeux différents, et doit donc pouvoir répondre à la quasi totalité des besoins en la matière. Il tourne sous Unix, mais des versions DOS et OS/2 sont également disponibles.

```
gide: /encode$ cat frompc.txt
Je vous confirme l'enregistrement de votre socit  notre sminaire technique du 15
juillet.
Merci et  bientt.
gide: /encode$ recode mac:latin1 frompc.txt
gide: /encode$ cat frompc.txt
Je vous confirme l'enregistrement de votre société à notre séminaire technique du 15
juillet.
Merci et à bientôt.
```

Figure 7.5 *Utilisation de la commande recode pour changer de jeu de caractère*

7.6 Configuration des types MIME et des polices de caractères dans les clients (email ou web)

On l'aura compris, un type MIME correspond à peu de choses près à l'extension d'un nom de fichier. Dans la pratique, c'est par cette même correspondance que les applications déterminent sous quel type MIME présenter tel fichier.

Cette section décrit la configuration de Netscape Mail pour ce qui concerne les types MIME, qu'il s'agisse d'envoyer ou de recevoir un document. La configuration d'Eudora Pro est tout à fait similaire.

Lorsqu'on souhaite envoyer un document (par exemple une image, un fichier GIF, disons `maison.gif`) par *email*, le programme chargé de l'encapsulation va lire le fichier, éventuellement l'encoder (par exemple en base64), et l'envoyer après avoir choisi le type MIME adéquat d'après l'extension.

À l'inverse, à la réception, le client va déterminer quel usage réserver au fichier selon le type MIME. Les tables qui associent les extensions de fichiers aux types MIME, et les types MIME aux actions à mener, sont contenues dans des fichiers de configuration.

7.6.1 Configuration des types MIME sous Netscape

Sous Netscape 2.x (version anglaise), les types sont configurables directement par l'intermédiaire du menu ***Options/Preferences***, dans l'onglet `Helpers`.

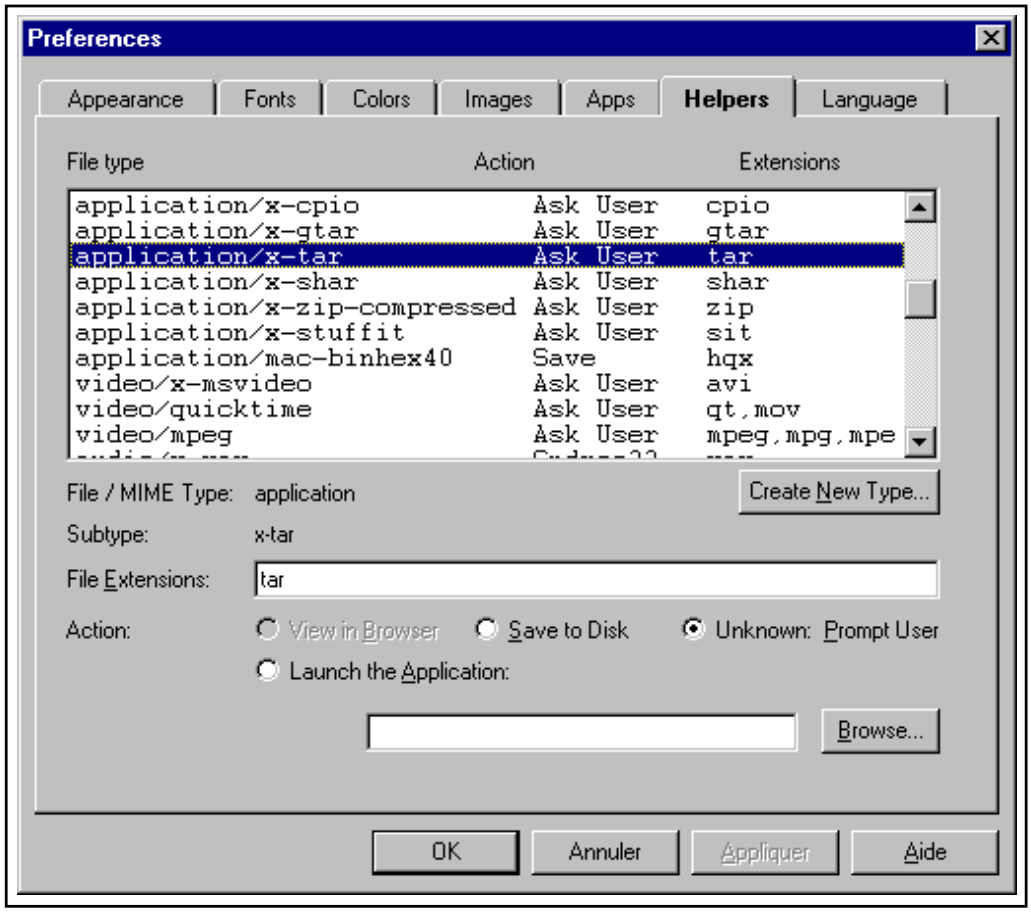

Figure 7.6 *Configuration des types MIME sous Netscape Navigator*

On y retrouve les types/sous-types, les actions à déclencher pour chacun de ces types, et les extensions des fichiers.

Quatre actions sont possibles :

– `View in Browser`, qui n'est disponible que pour les types réellement affichables dans la fenêtre du navigateur, c'est-à-dire le texte, les documents HTML, et certains formats d'images.

– `Save to Disk`, qui doit être choisie lorsqu'on souhaite conserver le fichier ou lorsqu'il n'est pas possible de l'envoyer directement à une application : par exemple un fichier compressé avec `gzip` devra être décompressé en ligne de commande.

– `Unknown: Prompt User`, le choix par défaut, qui déclenche l'affichage d'une boîte de dialogue pour demander à l'utilisateur ce qu'il souhaite faire du fichier (l'envoyer à une application, l'enregistrer sur disque...).

– `Launch the Application`, qui indique le programme à lancer pour traiter ou afficher le document. Par exemple, Netscape associe par défaut le type `audio/*` à l'application `naplayer` qui n'est autre qu'un utilitaire destiné à faire entendre le contenu d'un fichier son.

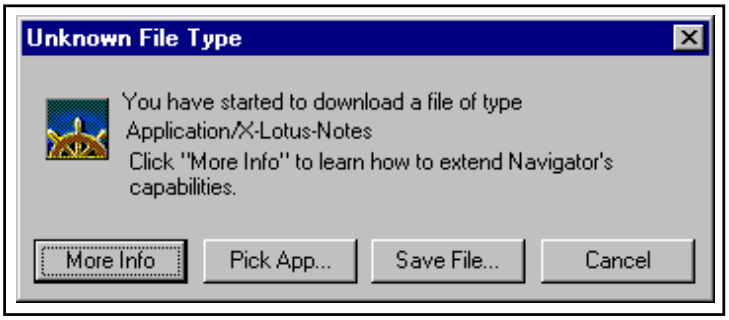

Figure 7.7 *Boîte de dialogue Netscape (type de fichier inconnu)*

Pour ajouter un nouveau type MIME, il suffit de cliquer sur le bouton `Create New Type`. Une fenêtre permet de saisir le type et le sous-type.

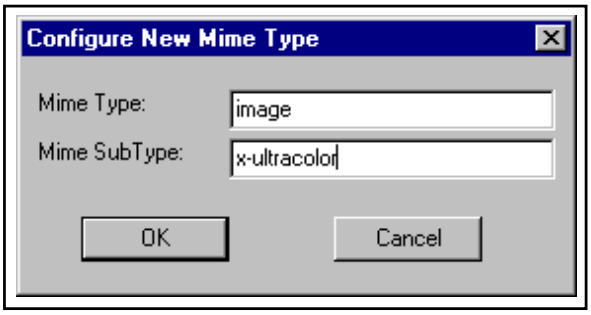

Figure 7.8 *Ajout d'un nouveau type MIME sous Netscape Navigator*

7.6.2 Configuration des polices sous Netscape

L'onglet `Fonts` du menu ***Options/Preferences*** de Netscape permet de spécifier les jeux de caractères à utiliser à l'affichage, tant pour les messages (*email*, *news*) que pour le Web. Il est utilisé conjointement avec les entrées du menu ***Options/Document Encoding*** qui permettent de changer de jeu de caractères.

Ainsi, pour visualiser un document rédigé dans un jeu de caractères Coréen il faudra sélectionner ce jeu de caractères (***Options/Document Encoding/Korean***) puis spécifier une police de caractères appropriée dans le menu `Fonts`. Deux formats de polices peuvent être définis : les caractères à espacement proportionnel (le texte courant) et les caractères à espacement fixe (utilisés pour les champs dans les formulaires, ou au sein des pages HTML pour mettre en évidence certains paragraphes particuliers, par exemple des extraits de code informatique). Bien entendu, il est indispensable de se procurer des polices Coréennes !

Plus simplement, la figure 7.11 page 254 montre un exemple de page affichée avec un jeu de caractères Grec. L'encodage Greek n'étant pas défini dans le menu ***Fonts***, nous utilisons le type `User Define` pour y associer la police adéquate (en l'occurrence ici une police Symbol), puis nous sélectionnons ce type pour l'affichage.

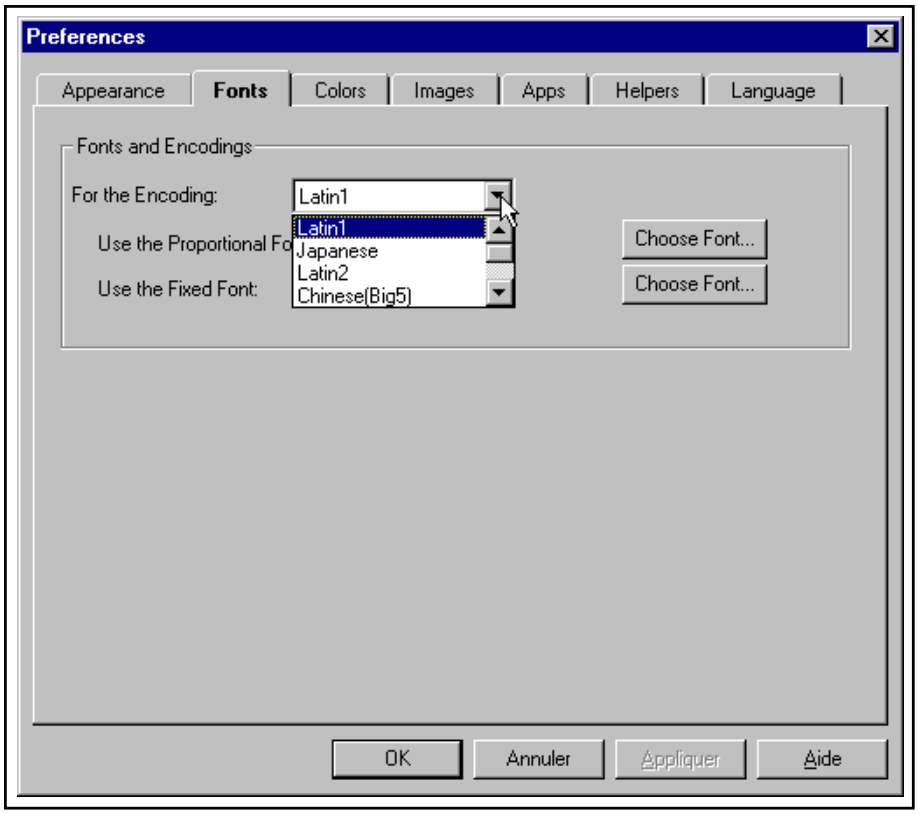

Figure 7.9 *Onglet de configuration des polices de caractères sous Netscape Navigator*

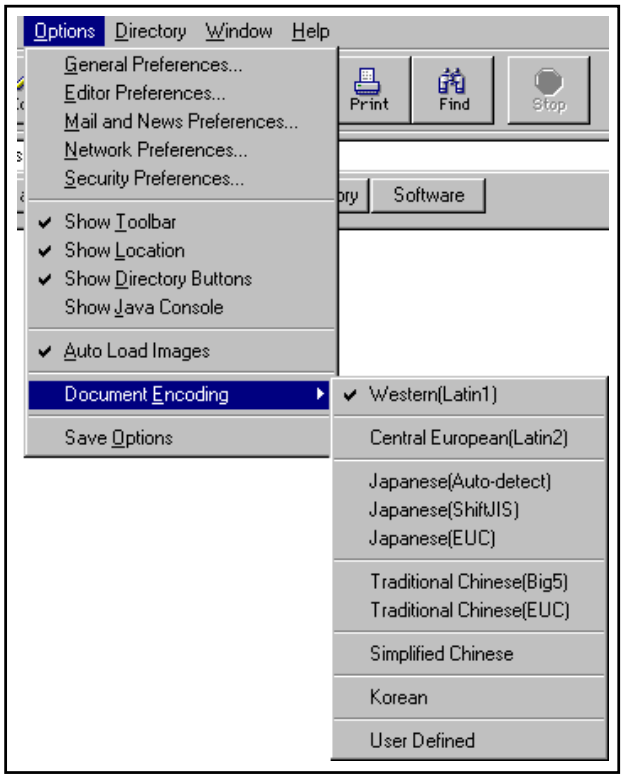

Figure 7.10 *Configuration du type d'encodage sous Netscape Navigator*

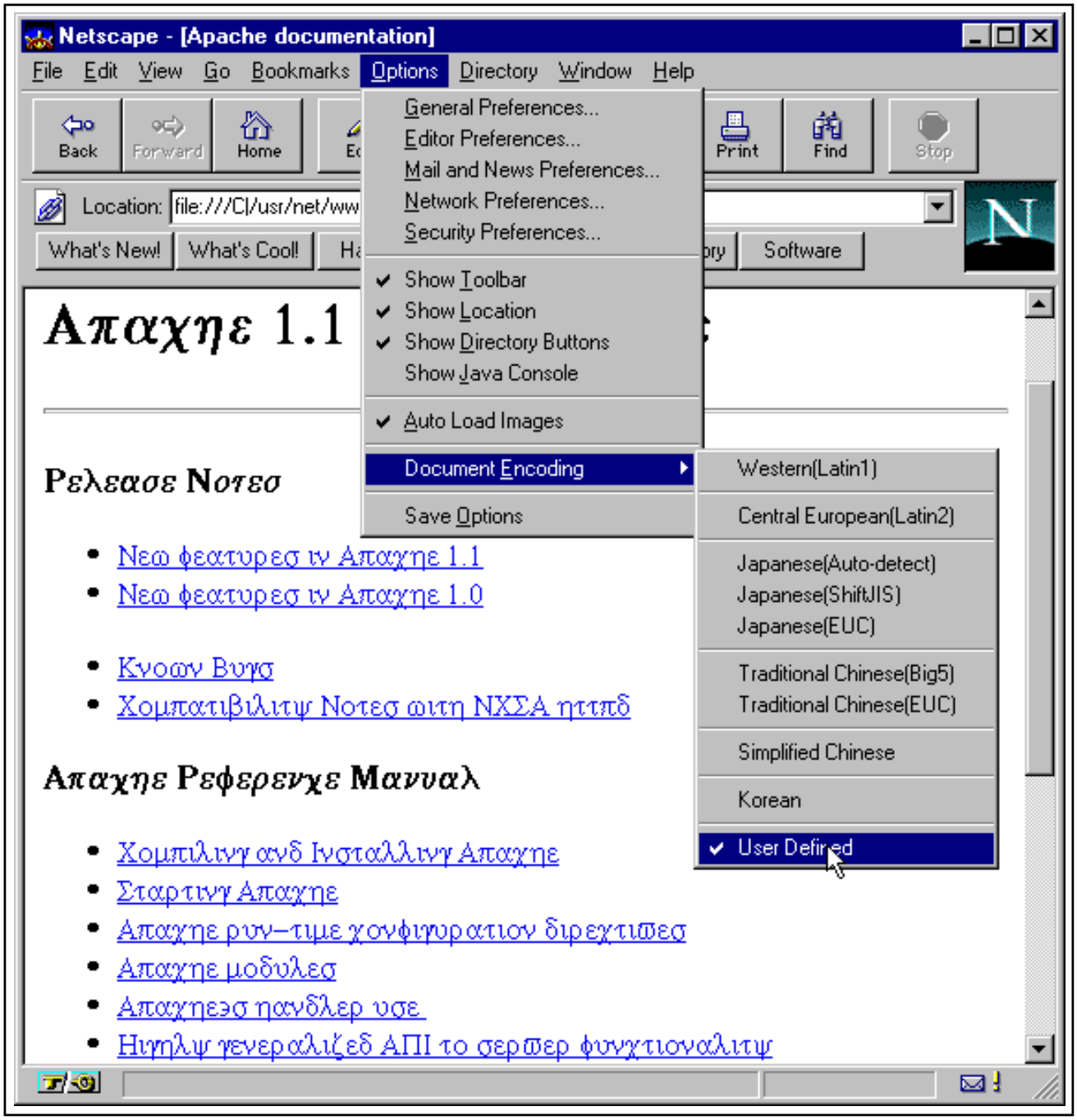

Figure 7.11 *Exemple de page affichée dans un jeu de caractères spécial*

8

Le web

Parmi les systèmes d'information les plus répandus sur l'Internet, le Web se place désormais largement en tête. Grâce à son interface plus agréable et plus simple à utiliser que la majorité des autres outils réseau, il s'est hissé au rang de standard de la « communication multimédia online ».

Pour une entreprise qui souhaite utiliser l'Internet comme vecteur de communication, le serveur Web est une étape presque incontournable. Non qu'il soit indispensable pour véhiculer l'information propre à l'entreprise, qu'elle soit commerciale, institutionnelle ou technique, mais simplement parce que de plus en plus d'utilisateurs s'attendent tout naturellement à pouvoir consulter cette information par ce biais.

Même en France, où l'Internet n'a pas pris son essor aussi rapidement que dans d'autres pays, le Web est devenu un média à part entière avec lequel il faut compter... et sur lequel on peut compter dans les années à venir. Durant le premier semestre 1996, le nombre de serveurs Web en France a doublé. Les premières applications de commerce électronique se sont mises en place. Selon les experts, 1997 sera une année charnière durant laquelle la communication multimédia sur l'Internet atteindra sa maturité et constituera une opportunité commerciale et professionnelle réelle.

8.1 Généralités

Le Web est un système d'information multimédia fondé sur un protocole de niveau applicatif (HTTP, HyperText Transfer Protocol) et un standard de formatage (HTML, HyperText

Markup Language). Ses principales caractéristiques sont les suivantes.

– Une délocalisation totale du contenu, puisque les documents sont accessibles indifféremment depuis n'importe quel nœud du réseau.

– La possibilité de faire référence à n'importe quel objet ou document au sein d'un autre document (c'est la notion de *lien hypertexte* ou *hypermédia*).

– La capacité à acheminer l'information vers des applications externes, ce qui en fait un outil complètement extensible.

On peut comparer l'ensemble du Web à une gigantesque base de données internationale, distribuée sur plus de 300 000 machines, dans laquelle il serait possible de mettre tous les documents en relation les uns avec les autres.

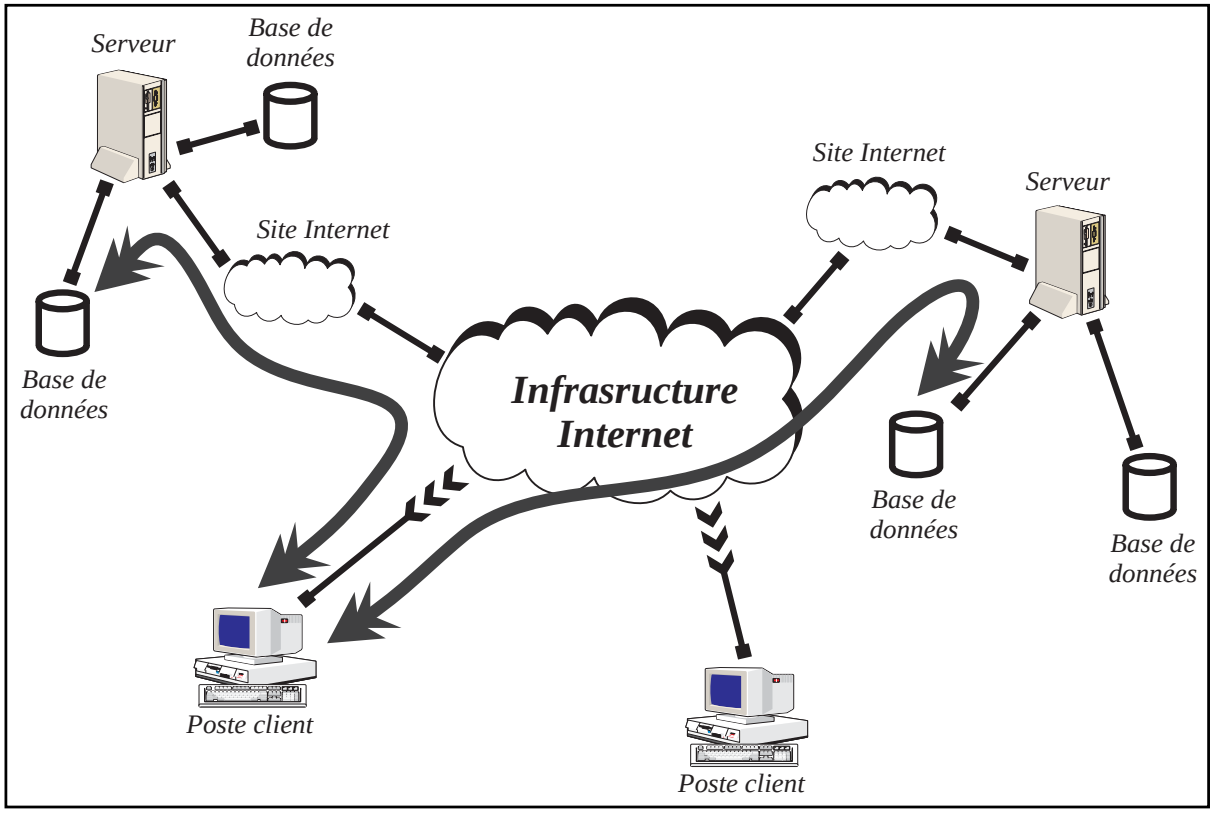

Figure 8.1 *Illustration de la structure distribuée du Web*

Cette notion d'hypermédia s'appuie sur un concept très simple permettant de localiser au sein de la base un document, ou plus généralement un objet informatique – une *ressource* – de manière non équivoque. On utilise pour cela la structure de l'Internet qui impose à chaque machine de disposer d'une adresse et éventuellement d'un nom univoque. La notion familière de *répertoire* peut ainsi être étendue à l'ensemble du réseau. Chaque ressource dispose donc d'un « chemin d'accès » unique : on parle d'*URL* (Uniform – ou Universal – Resource Locator).

Muni de cette possibilité de repérer n'importe quel document accessible sur le Web (voire sur n'importe quel autre service d'information ou de transfert de fichiers de l'Internet), on peut façonner des documents complexes, comprenant un grand nombre de pages, à partir de plusieurs documents physiquement situés sur des machines séparées par des milliers de kilomètres. C'est cette généralisation du lien hypertexte qui a fait la force du Web, qui est devenu par essence un lieu de coopération et d'échange.

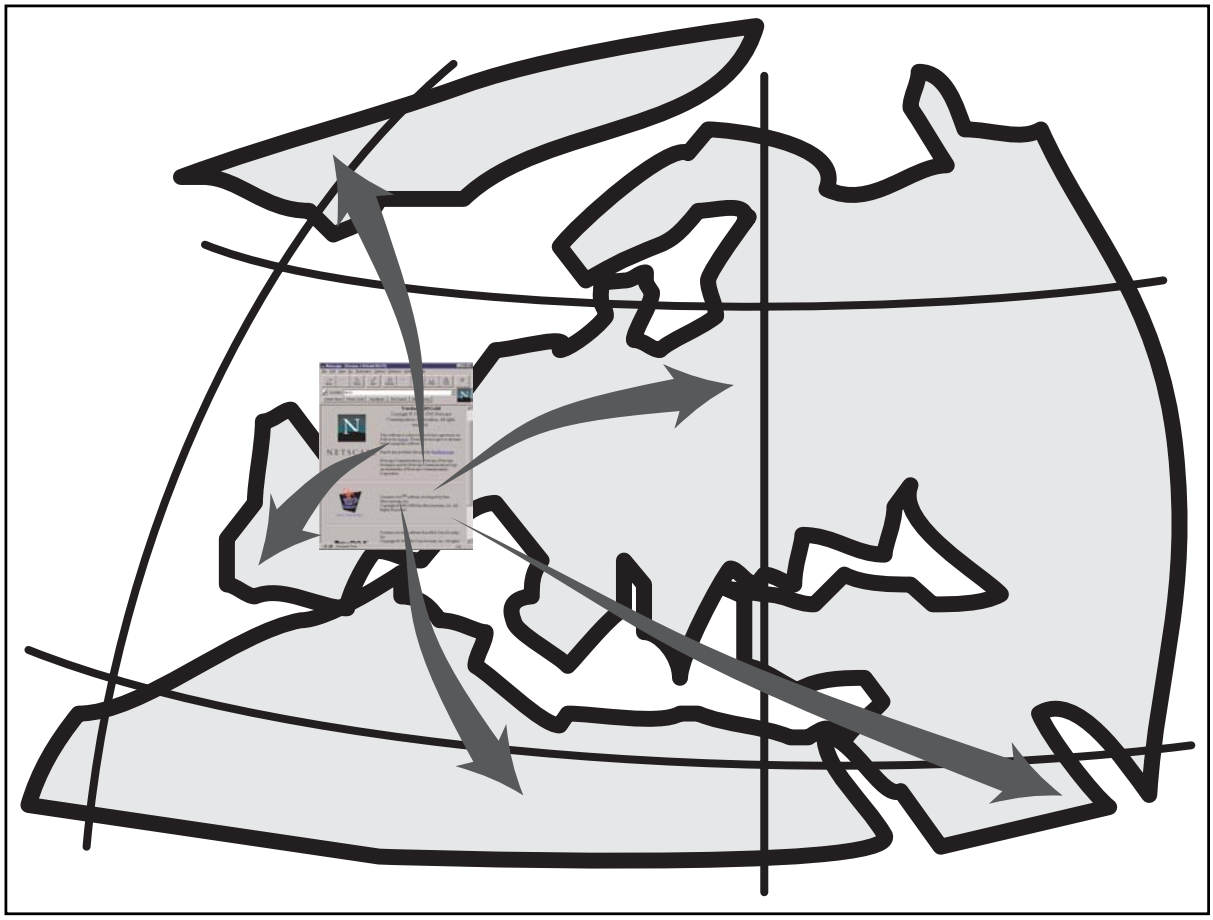

Figure 8.2 *Liens hypertexte à partir d'une page Web*

8.2 Les URL

La notion d'URL fait référence à une séquence de caractères permettant de retrouver n'importe quel type de ressource (fichier, document, message...) sur l'Internet. La syntaxe générale de cette séquence est de la forme :

```
methode://utilisateur:mot_de_passe@adresse:port/chemin
```

– `methode` indique le moyen d'accéder à la ressource (le protocole utilisé) ;

- `utilisateur` est le code d'accès utilisé pour se *connecter* au service désigné par le protocole ;
- `adresse` l'adresse IP ou le nom de machine (*host*) à interroger ;
- `port` le numéro de port UDP ou TCP ;
- `chemin` le nom complet du document, répertoire inclus. L'utilisateur et le mot de passe sont facultatifs, ils sont surtout utiles pour l'accès à des serveurs FTP ou à des serveurs Web sécurisés.

Exemples :

```
ftp://ftp.ibp.fr/pub/gnu/recode-3.4.tar.gz
http://www.eunet.fr/index.html
ftp://rousseau:rikiki03@opaque.fenetre.fr/Mail/
```

Cette dernière forme donne accès par FTP à la liste des fichiers du répertoire `Mail/` du compte de l'utilisateur `rousseau` sur la machine `opaque`. Pour le protocole FTP, si aucun code d'accès n'est précisé, la connexion se fait en FTP anonyme. Le client envoie alors un mot de passe quelconque, soit le nom de l'utilisateur, soit un nom choisi par les concepteurs du programme. Par exemple, Netscape Navigator envoie le mot de passe `mozilla@`.

Certains sites filtrent l'accès en FTP anonyme pour interdire les demandes formulées avec le mot de passe envoyé par Netscape Navigator ou d'autres applications comparables. On peut alors penser à un problème technique ; il n'en est rien, puisqu'il suffit de contourner le problème en indiquant un mot de passe *valable*. On pourrait imaginer par exemple :

```
ftp://anonymous:rousseau@fenetre.fr@ftp.ibp.fr/README
```

Manifestement, cet URL pose un problème, puisqu'il contient deux fois le caractère `@` : une première fois dans l'adresse *email* `rousseau@fenetre.fr`, une seconde fois pour indiquer le nom du serveur. Il est vraisemblable que le client enverra sa requête à la machine `fenetre.fr@ftp.ibp.fr`, qui bien entendu n'existe pas.

Cette remarque nous amène à présenter encore un procédé d'encodage, cette fois-ci propre aux URL.

8.2.1 Encodage des URL

Les caractères *illégaux* au sein d'un URL sont les suivants.

- Les caractères non affichables sur un terminal standard :
  - les codes de contrôle (code ASCII de 00 à 31 ou 0x1F) ;
  - les caractères propres au jeu ASCII étendu (code ASCII supérieur à 128 ou 0x80).

- Les caractères qui risqueraient d'induire une confusion dans le format, soit parce qu'ils servent de délimiteurs, soit parce que certaines applications les interprètent d'une manière spéciale.
- Le caractère % a un statut particulier, puisqu'à l'instar du signe = dans le standard MIME, il sert à indiquer un caractère encodé sous forme hexadécimale.

En résumé, les seuls caractères qui peuvent passer « en clair » sont les caractères alphanumériques (`a` à `z`, `A` à `Z` et les chiffres), plus les signes suivants :

```
$ - _ . + ! * ' ( ) ,
```

Bien entendu, il n'est pas interdit d'encoder tout de même ces caractères, au détriment de la lisibilité et de la concision de l'URL.

Le standard MIME représente les caractères encodés par le signe = suivi de leur code hexadécimal. Pour les URL, c'est le symbole % qui est utilisé. Ainsi, le signe @ qui nous gênait dans l'exemple cité plus haut peut s'écrire %40. Le symbole % lui-même sera représenté par %25.

Notre URL devient :

```
ftp://anonymous:rousseau%40fenetre.fr@ftp.ibp.fr/README
```

8.2.2 Les méthodes d'accès des URL

Les protocoles les plus couramment rencontrés dans les URL sont les protocoles de transfert ou de consultation : HTTP, gopher, FTP.

Il faut ajouter à cette liste certaines méthodes particulières.

- `file` donne accès à un fichier situé sur un disque de la machine de l'utilisateur. Cette méthode n'est pas suivie d'un nom de machine, mais uniquement du nom de fichier. Sous DOS/Windows, la lettre de lecteur est suivie du caractère `|` qui remplace le traditionnel signe ':'.

  Par exemple : `file:///C|/usr/net/gagnez_un_tshirt.htm`

  La troisième barre oblique (/) indiquée ici peut surprendre ; il faut en fait la considérer indépendamment des deux autres qui la précèdent puisque, comme sous Unix, elle fait référence à la racine du système de fichiers. Notons que les séparateurs de chemin sont représentés selon la syntaxe Unix, et non par l'habituel barre oblique inverse (\) de DOS ou Windows.
- `mailto` est suivi d'une adresse de courrier électronique. Ici aussi le format du reste de l'URL est particulier, puisque seuls le nom de l'utilisateur et le nom de domaine sont conservés.

  Par exemple : `mailto://luc@fenetre.fr`

– `telnet` est suivi uniquement du nom de la machine à contacter (et éventuellement d'un numéro de port).

  Certains navigateurs Web ajoutent à cette liste la méthode `viewsource` qui permet de consulter le source HTML ou plein texte d'un document sans aucune mise en page.

8.3 Les clients

8.3.1 Installation

Tous les exemples de ce chapitre exploitant les possibilités de *Netscape Navigator* dans sa version *3.0 Gold*, c'est cette version que nous nous proposons d'installer. L'installation des autres modèles de navigateurs Web est cependant tout à fait comparable. Notez que *Microsoft Explorer 3.0* pour *Windows 95* ou *Windows NT* présente la particularité de s'intégrer totalement au système puisqu'il dote l'explorateur (l'outil permettant d'afficher l'ensemble des ressources de la machine et d'explorer les répertoires des disques) de la capacité d'afficher des documents HTML.

Installation de Netscape sous Windows ou Macintosh

L'archive récupérée est dite *auto-extractible*, c'est-à-dire qu'il s'agit d'un exécutable capable de gérer entièrement l'installation. Il n'est pas nécessaire de recourir à un quelconque utilitaire complémentaire. Toute l'installation est pilotée très simplement par des menus lorsqu'on l'exécute.

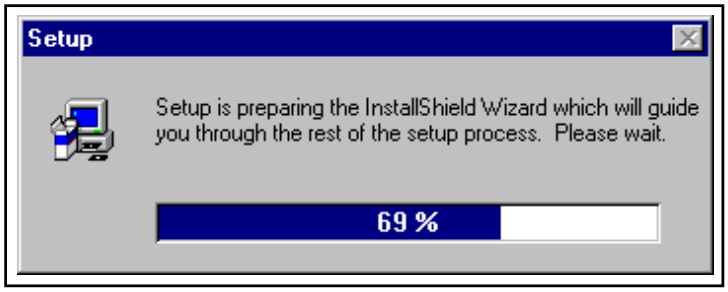

Figure 8.3 *Boîte de dialogue de l'utilitaire d'installation de Netscape Navigator*

Comme à l'accoutumée, il faut sélectionner un répertoire d'installation. Ici, il s'agit de la version de Netscape Gold pour Windows 95 ; le répertoire par défaut est :

```
\Program Files\Nav_Gold
```

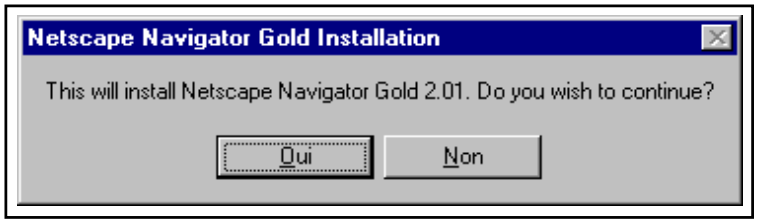

Figure 8.4 *Message de l'utilitaire d'installation de Netscape Navigator*

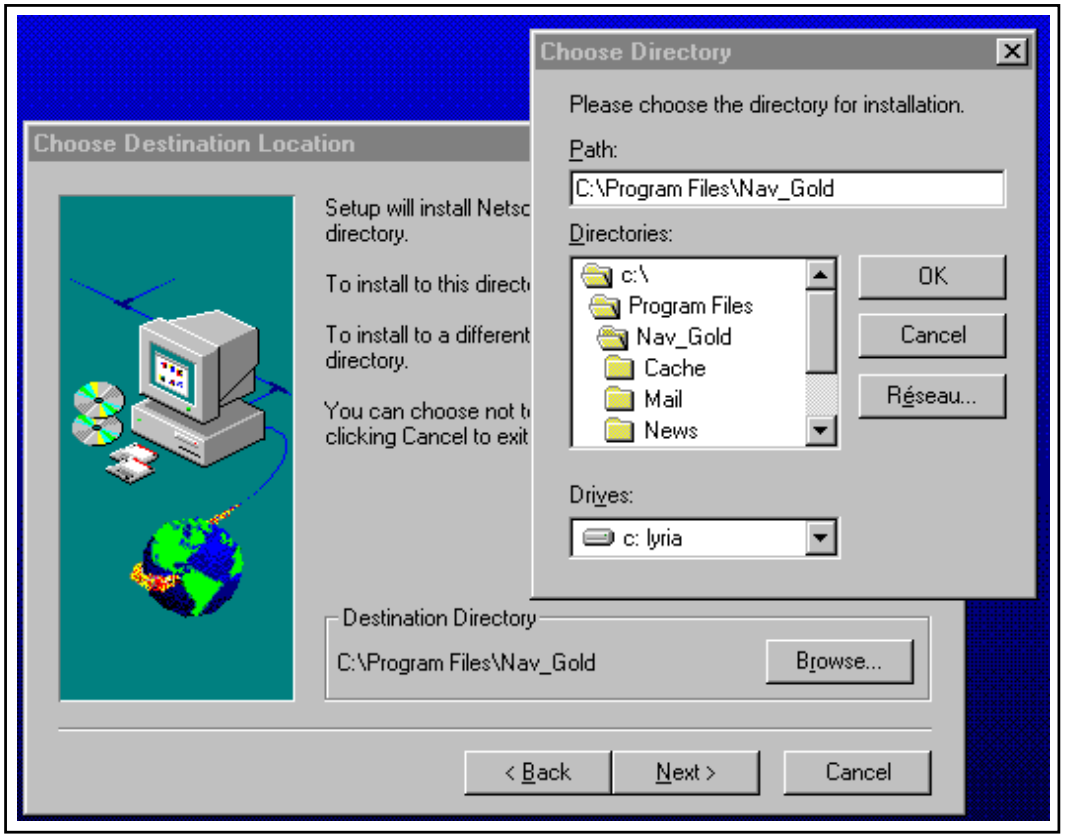

Figure 8.5 *Choix d'un répertoire d'installation pour Netscape Navigator*

On notera qu'il est possible d'installer sans difficulté une nouvelle version de Netscape Navigator par-dessus une ancienne. Les informations de configuration seront conservées, ainsi que les dossiers réservés au courrier électronique, aux signets (*bookmarks*), etc.

8.3.2 Configuration

Présentation générale de l'application Netscape Navigator

La fenêtre principale du navigateur (*browser*) comporte par défaut une barre de menus et une barre de navigation en haut (figure 8.6 page suivante), et une barre d'outils en bas. L'espace du

milieu est réservé à l'affichage des documents HTML ou texte. Par défaut après l'installation, le premier document affiché est la page d'accueil du site Web de Netscape.

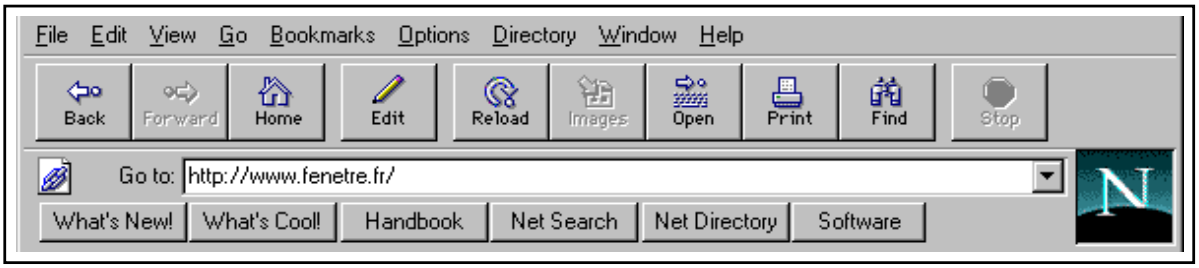

Figure 8.6 *Barres de menu et de navigation de Netscape Navigator*

La barre de navigation permet de se *déplacer* sur le Web, de charger les images d'un document, d'imprimer, etc. Un champ (`Location:`) est réservé pour entrer l'URL qu'on souhaite visiter.

Une rangée de boutons donne accès à quelques ressources utiles sur le site Web de Netscape Corp. : les nouveautés et les produits de Netscape, un annuaire de recherche (***NetSearch***)...

Le menu ***File*** présente toutes les options nécessaires pour ouvrir un URL ou un fichier local (comme nous l'avons déjà mentionné, ce type de fichier peut être représenté par un URL dont la méthode d'accès serait `file`), pour enregistrer sur disque une page HTML affichée à l'écran. Dans la version Gold de Netscape Navigator, qui intègre un éditeur HTML, ce menu offre également la possibilité d'éditer l'URL en cours ou de créer une nouvelle page.

Figure 8.7 *Ouverture d'un URL ou d'un fichier dans Netscape Navigator*

Le menu ***Go*** joue le même rôle que la barre de navigation, puisque outre les options `Back` et

`Next`, il rappelle l'historique des URL visités.

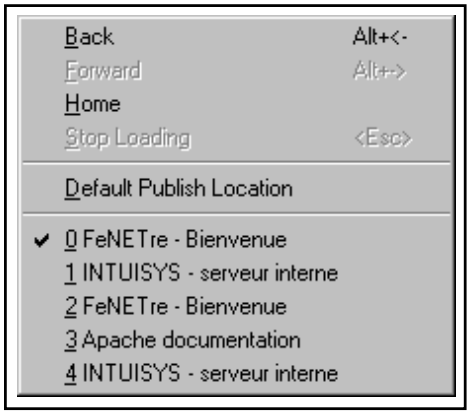

Figure 8.8 *Historique des URL sous Netscape Navigator*

Le menu ***Bookmarks*** donne accès, comme son nom l'indique, aux signets. Le principe de cette liste d'URL est simple, puisqu'il s'agit d'une sorte de bibliothèque de références qu'on peut enrichir à loisir. L'option `Add Bookmark` ajoute à la liste la référence de l'URL en cours, tandis que l'option `Go to Bookmarks` affiche une fenêtre qui permet de mieux organiser ses liens, comme sous un gestionnaire de fichiers (la notion de dossier y est d'ailleurs présente).

Sous ces options, la liste des URL disponibles nous permet d'un simple clic de la souris de retrouver un site qui nous aurait semblé intéressant et que nous aurions ajouté à notre liste.

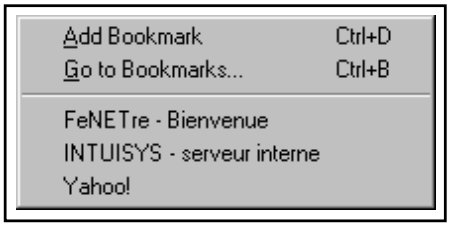

Figure 8.9 *Les signets (Bookmarks)*

Maintenant que l'aspect général de l'application nous est un peu plus familier, nous allons étudier les options de configuration avant d'évoquer la navigation proprement dite. La création de pages HTML ne relève pas de cette partie ; elle sera vue au chapitre 10 page 319.

Configurer Netscape pour les différentes méthodes d'accès URL

La fenêtre de configuration des applications, accessible par le menu ***Options/General Preferences*** dans l'onglet `Apps`, permet d'indiquer quelles applications doivent être lancées lorsqu'un URL concerne le protocole `telnet`. La ligne fait référence aux connexions `telnet` vers des *mainframes* IBM. Il est également possible d'indiquer quelle application utiliser pour

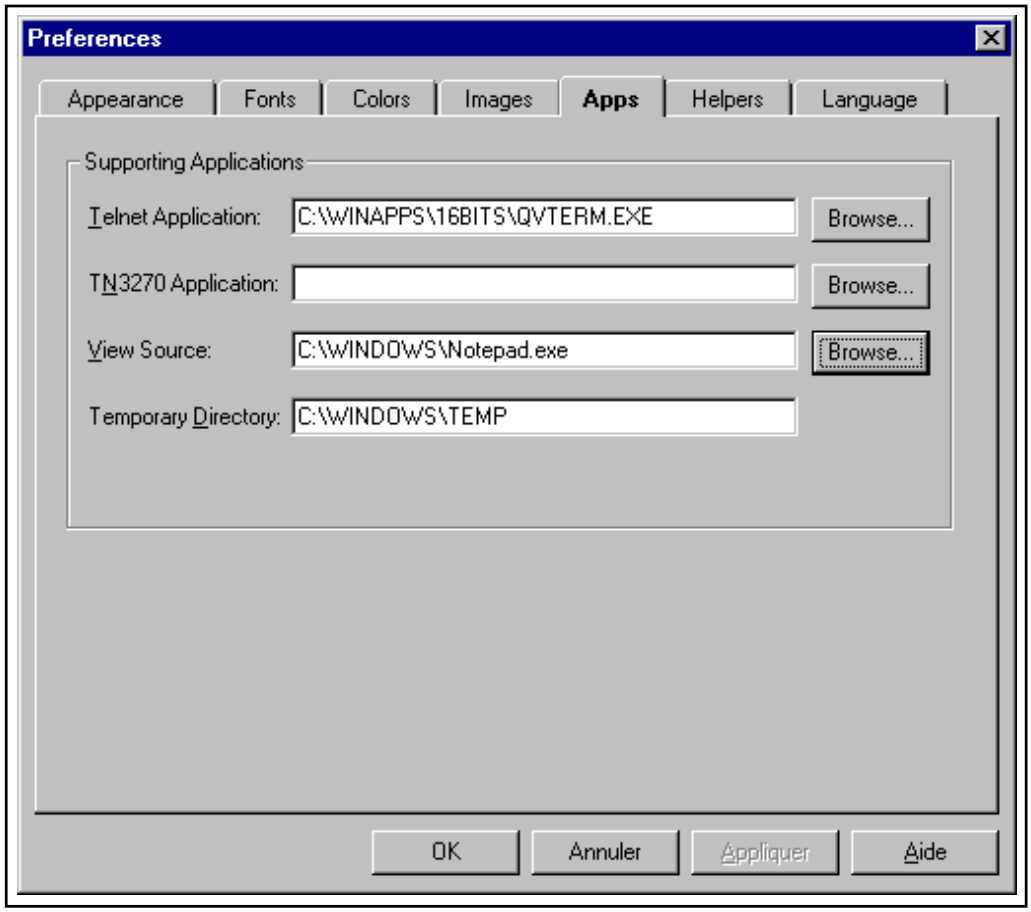

Figure 8.10 *Configuration des méthodes d'accès aux URL sous Netscape Navigator*

afficher (ou éditer) le source des documents HTML. Par défaut c'est l'utilitaire intégré dans Netscape qui s'en charge, mais il ne permet pas d'éditer ni de sauvegarder les documents : il peut donc être utile de le remplacer par un éditeur HTML ou un éditeur de texte.

La boîte de dialogue General Preferences

Nous avons déjà décrit à la section 7.1.1 page 240 les onglets `Fonts` et `Helpers` de la boîte de dialogue ***Options/General Preferences***, et l'onglet `Apps` au paragraphe précédent.

Voyons dans l'ordre les onglets restants.

L'onglet `Appearance` permet de définir le comportement et la présentation de l'application elle-même. On peut ainsi choisir d'afficher la barre de menus avec ou sans les icones ou le texte. En indiquant quelles applications lancer au démarrage, on peut en outre automatiser l'affichage du courrier électronique ou des forums lors de l'utilisation du navigateur.

Au lancement de Netscape Navigator, l'application va par défaut charger l'URL du site Netscape. Pour changer ce comportement, on peut demander à ce qu'aucun URL ne soit chargé (`Start With Blank Page`) ou bien indiquer un autre site. Ici, c'est la page d'accueil du serveur de la société FeNETre qui est sélectionnée ; elle apparaîtra donc automatiquement

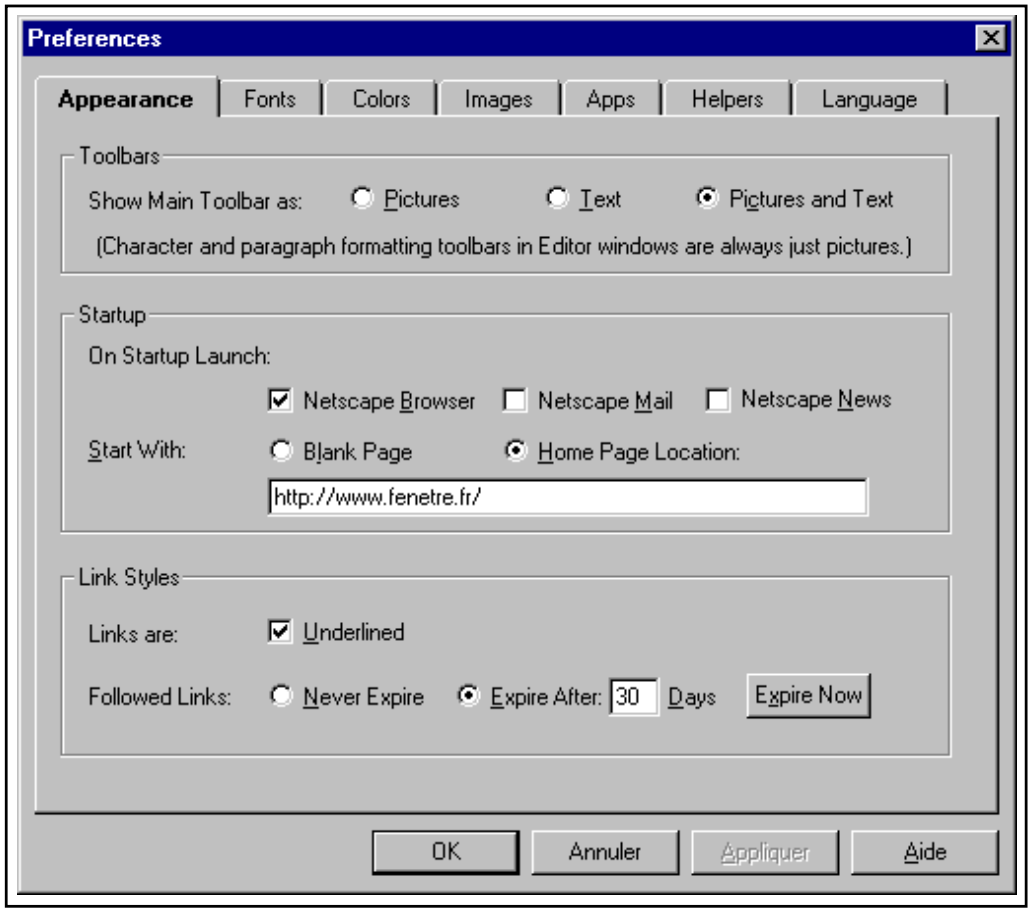

Figure 8.11 *Configuration de la fenêtre de Netscape Navigator*

(à condition que le serveur soit accessible, bien entendu).

L'onglet `Colors` offre la possibilité de changer les couleurs par défaut utilisées dans les documents affichés. Nous reviendrons sur la notion de lien hypertexte, retenons simplement que c'est dans cette boîte de dialogue qu'on peut choisir la couleur du texte, la couleur ou le motif de fond, et la couleur des liens, déjà visités (`Followed Links`) ou non.

Pour ce qui concerne le fond du document, on peut indiquer une couleur (pour Netscape sous Windows, la couleur par défaut est le blanc ; d'autres navigateurs, surtout les moins récents, utilisent le gris clair) ou une image. Lorsqu'ils doivent afficher une image en fond, les navigateurs l'utilisent comme un motif de remplissage : c'est-à-dire que partant du coin haut gauche, ils répètent cette image vers la droite et vers le bas autant de fois qu'il est nécessaire pour remplir tout l'espace d'affichage.

On remarquera l'option `Always Use My Colors`. Elle permet d'imposer que les couleurs choisies soient utilisées partout. Nous verrons lorsque nous évoquerons la création de pages HTML qu'il est possible de définir ces mêmes couleurs pour chaque document. C'est ainsi que certains sites apparaissent avec un fond bleu, d'autres avec un motif jaune. Si, pour des raisons de lisibilité, on préfère définir ses propres couleurs, il suffira de cocher cette case.

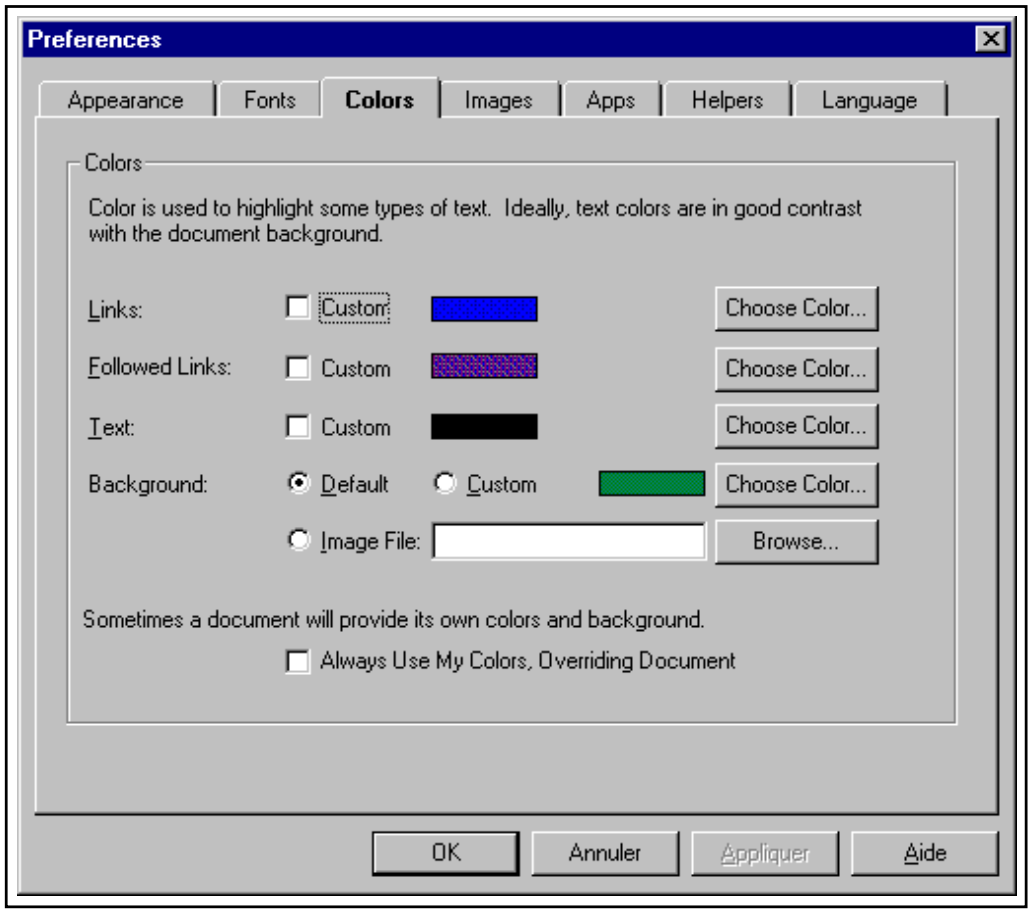

Figure 8.12 *Choix des couleurs d'affichage sous Netscape Navigator*

Figure 8.13 *Utilisation d'une image comme motif de fond de page d'un navigateur*

Cependant, certains sites perdront de leur richesse et risqueront de paraître ternes.

L'onglet `Image` se rapporte également à l'affichage des couleurs (`Choosing Colors`). On peut en outre décider de n'afficher les images (`Display Images`) que lorsqu'elles ont été entièrement chargées, ou bien de les afficher au fur et à mesure.

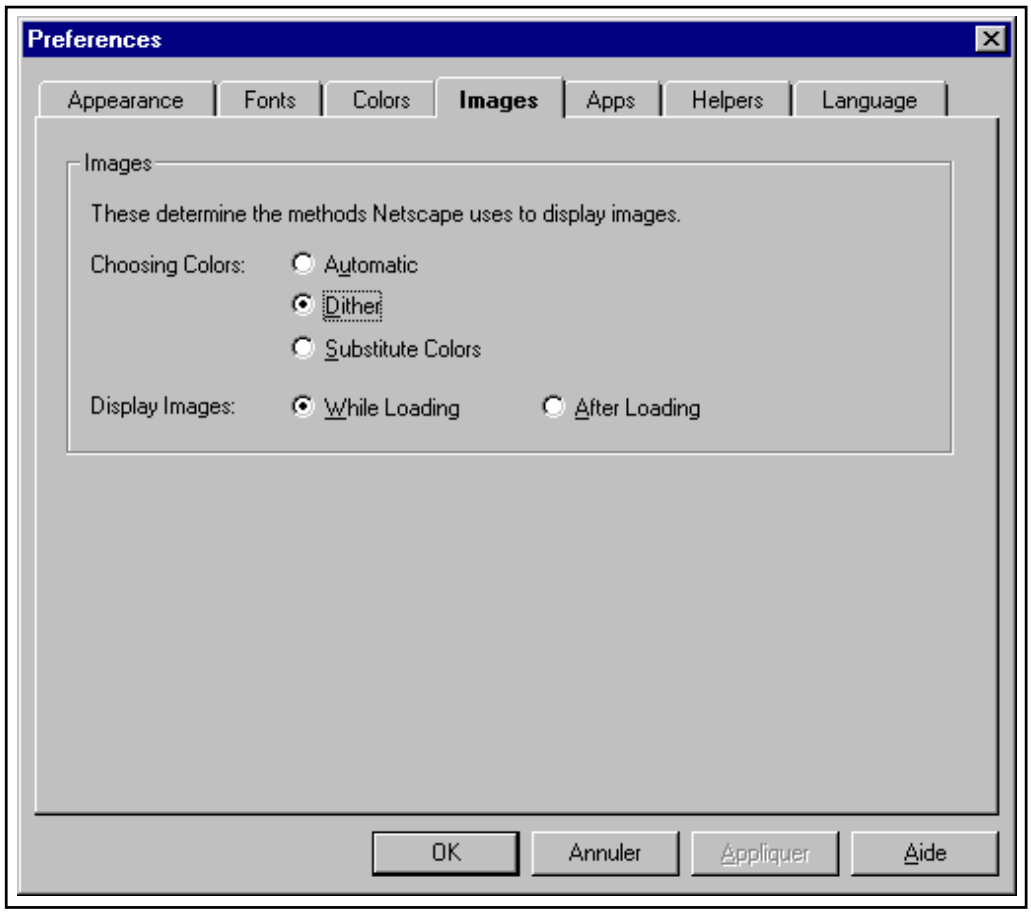

Figure 8.14 *Méthodes de réduction des couleurs*

On retiendra que selon les sites visités (et en particulier selon les outils utilisés par les concepteurs de ces sites pour fabriquer les images) les gammes de couleurs (on parle de *palettes*) peuvent varier. Puisque Netscape les affiche avec un jeu de couleurs limité sur la plupart des systèmes (en mode 256 couleurs sous Windows, il est en fait figé), il peut être nécessaire de choisir quel type d'approximation appliquer. La technique dite du *dithering* diffuse les couleurs les unes dans les autres, en utilisant des trames de pixels répartis de manière aléatoire. Elle permet d'obtenir un rendu plus réaliste, mais avec des contours parfois plus flous.

Le dernier onglet, `Language`, concerne les langues acceptées lors d'une consultation. Son utilité est toute relative, car leur configuration se fonde sur une information particulière contenue dans l'en-tête des messages HTTP, or cette information n'est que très rarement prise en compte par les serveurs. En théorie, elle permet au serveur de nous envoyer une version d'un document dans une langue que nous pouvons lire. Les flèches permettent d'ajouter à la liste

des langues acceptées celles que comprend l'utilisateur.

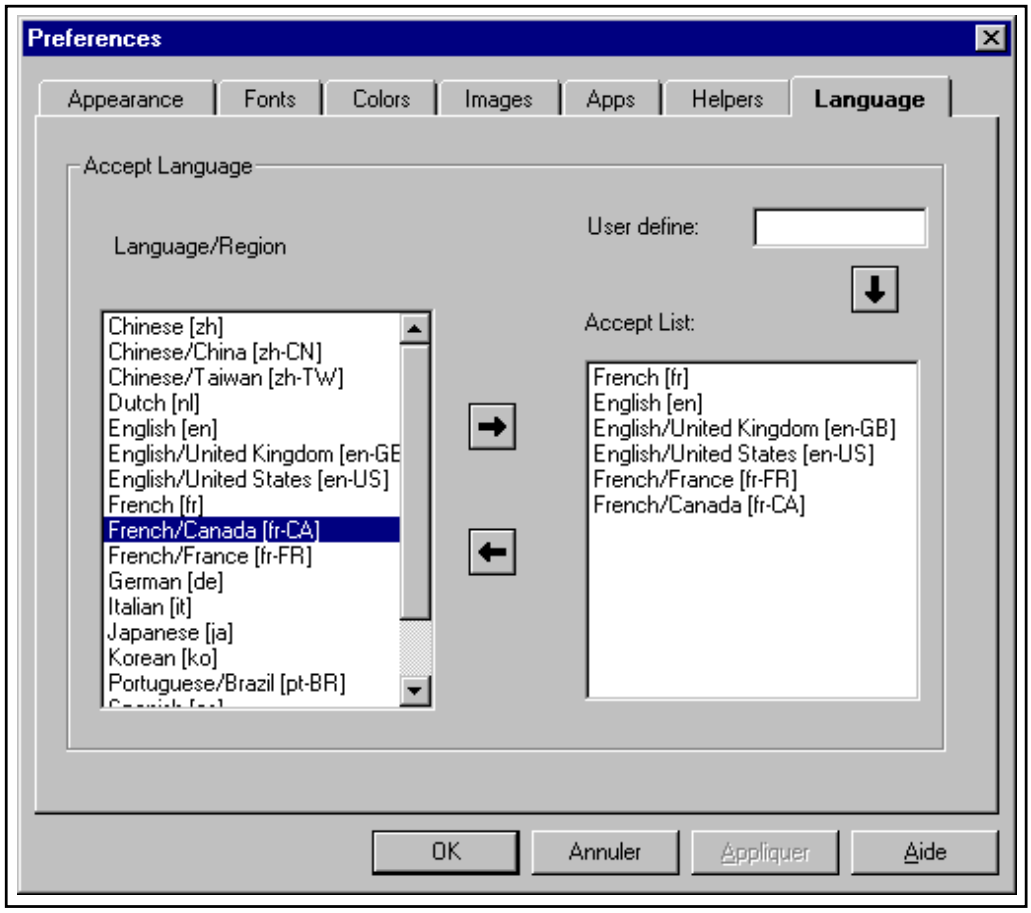

Figure 8.15 *Choix des langues acceptées par l'utilisateur*

La boîte de dialogue Network Preferences

Elle se compose de trois onglets. Le dernier concerne les serveurs *proxy* qui sont présentés au chapitre 11 page 371.

Lorsqu'on navigue sur le Web, il est fréquent de revenir régulièrement à certaines pages déjà chargées : par exemple, la page d'accueil du site auquel on s'intéresse risque fort d'être un point de passage obligé pour accéder aux différents menus.

Afin de limiter les délais liés au téléchargement des données, les navigateurs utilisent un *cache* pour conserver pendant un certain temps les informations qui reviennent le plus souvent. Bien sûr, lorsqu'il s'agit de pages HTML ou de documents texte, les temps de chargement sont suffisamment faibles pour qu'on puisse se dispenser de ce type de solution ; cependant le Web est un média à vocation universelle, et les images ou les vidéos qui y circulent représentent parfois de grands volumes de données.

L'onglet `Cache` des préférences de Netscape permet d'indiquer une taille maximale pour les deux niveaux de cache, mémoire et disque. Les informations sont d'abord conservées en mémoire, puis, lorsqu'il faut libérer de la place pour des informations plus récentes, transférées

sur disque où elles resteront jusqu'à ce qu'elles soient trop anciennes pour être conservées.

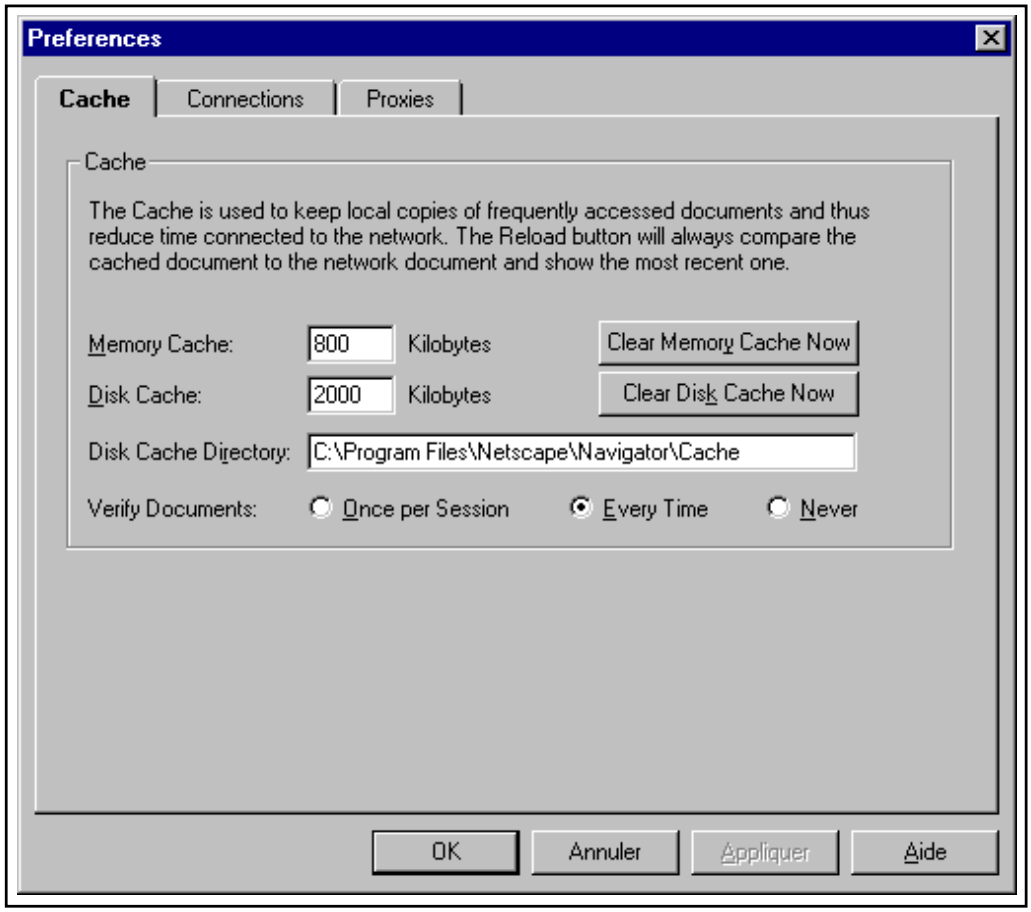

Figure 8.16 *Configuration du cache sous Netscape Navigator*

Les valeurs proposées par défaut sont convenables. On peut réduire la taille du cache mémoire sur une machine qui dispose de peu de mémoire vive. Cependant, si on supprime complètement les caches, certains effets particuliers, comme par exemple les animations GIF en boucle, ne se produiront plus. Il n'est généralement pas souhaitable de supprimer tout l'espace cache, sauf pour économiser à tout prix la mémoire et le disque sur une machine peu performante reliée à un *proxy-cache* qui jouera de toute façon ce rôle.

Comme nous le verrons, une *page HTML*, au sens de ce qui s'affiche à l'écran lorsqu'on demande le chargement d'un URL, est en fait composée de plusieurs documents/fichiers : le code HTML lui-même, plus les images et éventuellement les sons, vidéos, etc.

Pour accélérer l'apparition des données, il est possible de demander à Netscape de télécharger plusieurs fichiers à la fois (en réalité, il devra tout d'abord charger le code HTML, puisque c'est dans ce code que se trouvent les noms des fichiers – images par exemple – à charger). L'onglet `Connections` permet de préciser le nombre maximal de documents chargés simultanément.

Il est inutile d'indiquer un nombre trop grand, surtout si le débit de la connexion à l'Internet

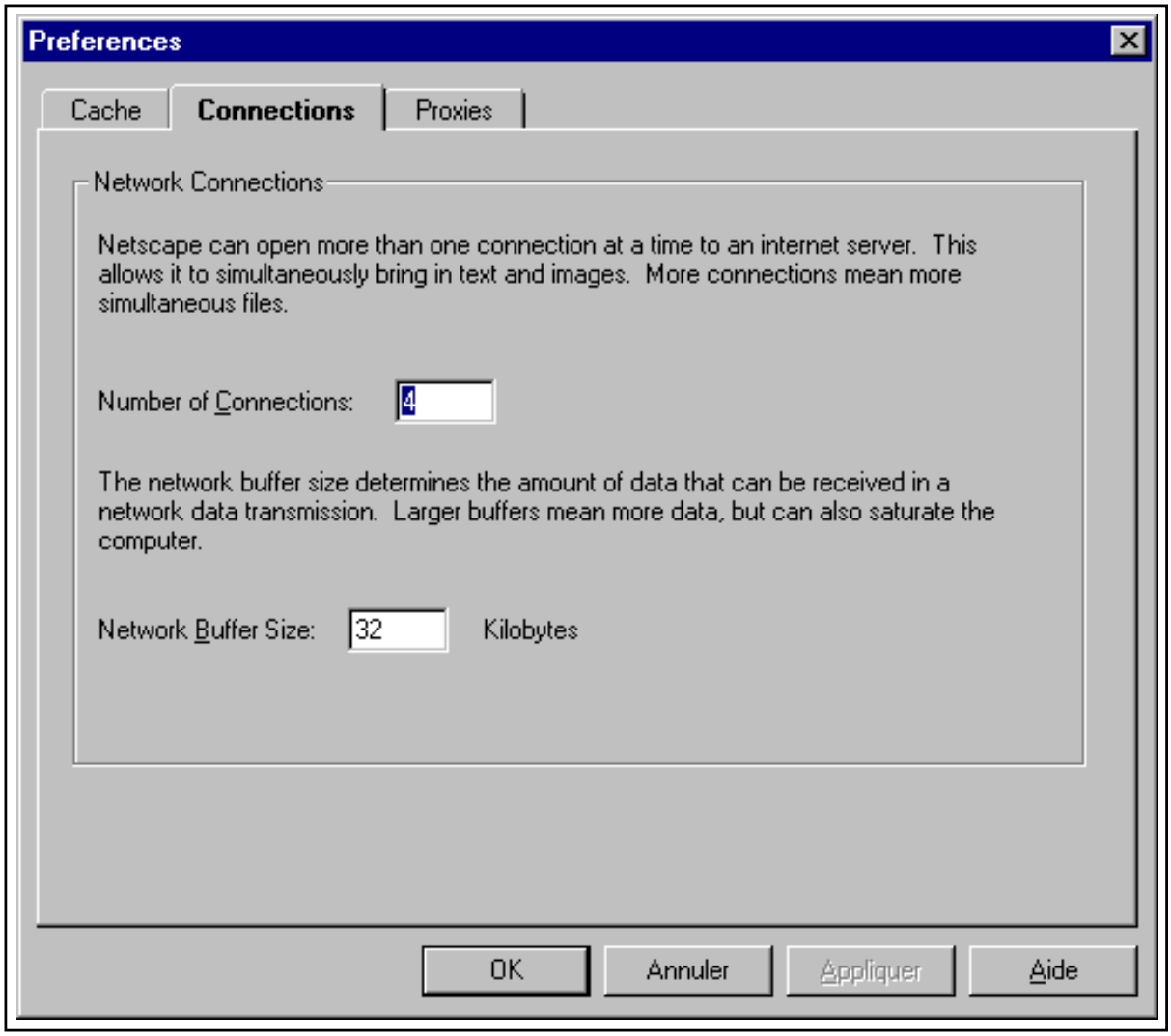

Figure 8.17 *Choix du nombre de connexions simultanées*

est faible : cela ne ferait qu'encombrer inutilement le réseau en générant des collisions qui d'une manière générale ralentiraient le chargement des pages.

Par ailleurs, il est rare qu'une page contienne plus de quatre ou cinq images. La valeur par défaut (4) devrait être conservée.

On peut également indiquer la taille des blocs de données à lire sur le réseau. Pour une machine peu puissante, il vaut mieux limiter cette valeur.

8.4 Naviguer sur le Web

Si on utilise souvent une terminologie empruntée au nautisme pour désigner le fait d'aller chercher de l'information sur le Web, c'est que de fait, trouver cette information relève parfois

d'un véritable casse-tête et nécessite de faire preuve de patience et d'un excellent sens de l'orientation, qualités précieuses chez un navigateur chevronné.

De par sa capacité à faire référence, au sein d'un même document, à plusieurs sources d'information provenant d'endroits physiquement et administrativement très éloignés, le Web ressemble à la toile d'araignée d'où il tire son nom.

L'information y est rarement séquentielle, parfois totalement distribuée, le plus souvent arborescente.

8.4.1 Structure des documents et des sites

Comme nous le verrons dans la section qui lui est consacrée, le langage HTML utilisé sur le Web pour formater les documents écrits repose sur quelques principes simples.

- *La généricité*, puisqu'un même document doit fournir la même information quel que soit l'endroit et le logiciel utilisés pour le consulter (cette règle est malheureusement de moins en moins respectée, les impératifs économiques poussant chaque éditeur de logiciel à n'en faire qu'à sa tête pour offrir plus de possibilités que ses concurrents).
- *La lisibilité*, c'est-à-dire qu'un document HTML doit être lisible par l'homme sans moyens autres qu'un quelconque éditeur de texte, et à ce titre ne peut contenir de caractères spéciaux (le jeu de caractères iso-latin1 est cependant accepté dans les nouvelles versions des navigateurs).
- *L'universalité*, ce qui signifie qu'un document peut faire référence à n'importe quel type d'information extérieure, et ce de n'importe quel endroit (ce qui est possible sur l'Internet, grâce aux URL).

Dans cette optique, un document consultable sur le Web est donc constitué d'un ou plusieurs fichiers rédigés en HTML (portant généralement l'extension `.html` sous Unix et `.htm` sous DOS ou Windows), et des fichiers qui l'accompagnent : images, graphiques, sons, etc.

On parlera souvent de *document* pour désigner l'équivalent d'un document écrit : ce peut être un simple fichier HTML, mais aussi bien une série de plusieurs fichiers liés entre eux, avec les éléments graphiques qui les accompagnent.

On parlera de *page* HTML pour désigner indifféremment un fichier HTML ou l'ensemble constitué par un fichier HTML et les éléments graphiques associés. La distinction entre les pages d'un document écrit et les pages d'un document HTML est difficile à établir, car si la structure du document papier est nécessairement linéaire, celle du document HTML ne l'est pas forcément. De plus à l'affichage, une page HTML peut représenter plusieurs écrans, c'est-à-dire qu'il peut être nécessaire d'utiliser l'ascenseur de la fenêtre du navigateur pour dérouler le texte.

Un navigateur doit tout d'abord se procurer le source de la page HTML avant de déterminer quels fichiers supplémentaires il doit télécharger. Quant à l'utilisateur, il n'a généralement

pas à demander le chargement d'une image ou d'un son, mais simplement de la page HTML qui y fait référence. Le format d'une requête – son URL – sera donc par exemple :

```
http://www.fenetre.fr/nouveautes/septembre/velux.html
```

Organisation des sites

Les sites Web sont le plus souvent arborescents. À partir d'une page d'accueil (*home page*), des liens permettent d'explorer telle ou telle branche du site. Ainsi le site Web de la société FeNETre s'articule-t-il autour d'une page d'accueil qui joue également le rôle de sommaire général, de six sommaires de *rubriques*, et pour chaque rubrique de quelques pages de contenu.

Pour des raisons de commodité, les concepteurs de sites Web calquent fréquemment l'arborescence physique des répertoires dans lesquels se trouvent les fichiers html sur l'arborescence logique du site.

Le Web a été conçu à l'origine pour permettre l'échange de documents, en supposant l'existence de fichiers et de répertoires. Même si la structure logique de l'information a de plus en plus tendance à masquer l'organisation réelle des fichiers sur les disques, les serveurs http sont capables de répondre à des requêtes portant sur des répertoires au lieu de porter sur des fichiers html. Le plus souvent, ils le font en expédiant un fichier au statut particulier, dit fichier d'index, et très souvent nommé `index.html`. Lorsque ce fichier (ou tout autre fichier auquel on aurait donné le statut de fichier d'index) n'existe pas, et si l'opération n'a pas été explicitement interdite lors de la configuration du serveur, c'est la liste des fichiers du répertoire qui est transmise. Le navigateur se comporte alors comme un gestionnaire de fichiers distants.

Pour accéder à la page d'accueil du site `www.fenetre.fr`, il suffira d'appeler l'URL suivant :

```
http://www.fenetre.fr/
```

en omettant de préciser le nom du fichier `index.html`, puisque le serveur ira automatiquement chercher ce fichier. D'ailleurs, sous Netscape, il n'est pas non plus indispensable d'indiquer la méthode d'accès : il s'agit par défaut du protocole http. L'URL à entrer dans le champ ***Open Location*** sera donc simplement `www.fenetre.fr`.

Comme on peut le voir, certains mots du texte de cette page apparaissent en caractères soulignés. Ce sont les liens hypertexte : cliquer dessus permet d'accéder aux pages qu'ils désignent. La barre de menus en bas de la fenêtre est également sensible à la souris : on peut établir des liens hypertexte à partir d'images aussi bien que de mots.

Notons que contrairement à certaines idées reçues, le nom d'un serveur Web ne commence pas nécessairement par *www* ; cette convention d'usage n'existe que dans un souci de discri-

Figure 8.18 *L'index d'un site est en fait sa page d'accueil*

mination des machines hébergeant tel ou tel service particulier (ainsi `ns` pour les serveurs de noms).

Le paragraphe suivant va nous donner l'occasion d'expérimenter ces notions de navigation hypermédia.

8.4.2 Rechercher l'information

Étant donnée l'étendue du Web, qui représente aujourd'hui plus de 50 millions de pages HTML réparties sur plus de 300 000 serveurs, il est virtuellement impossible de retrouver instantanément une information particulière, à moins de savoir très précisément où elle se trouve.

De nombreux annuaires et services de recherche sont disponibles pour tenter de localiser les documents. Ils permettent de chercher les sites d'entreprises à contacter, d'obtenir la liste des serveurs où l'on vend tel ou tel produit, de demander l'adresse d'un site annonçant les derniers résultats de la coupe du Monde de football, ou de tracer son chemin dans les archives des plus grands musées et bibliothèques.

On distingue principalement deux types de services : les annuaires, qui sont de simples collections de noms de sites et de sociétés, comparables aux pages jaunes de l'annuaire du téléphone, et qui sont le plus souvent maintenus à jour par les organismes qui administrent l'Internet, et les moteurs de recherche, sortes de bases de données géantes qui explorent quotidiennement le Web et archivent tout ce qu'ils y trouvent pour ensuite le restituer à partir d'une liste de mots clés ou en recherchant un mot, une phrase qu'on leur soumet.

Les applications et l'utilité des uns et des autres sont différentes. Autant les moteurs de recherche sont indispensables pour retrouver une information quelconque parmi les milliards de lignes de texte que représente le Web, autant les annuaires constituent le moyen le plus simple de retrouver une entreprise et son serveur Web si elle en dispose sans devoir passer en revue des centaines, voire des milliers de pages.

Les annuaires

Les annuaires recensent les adresses des sites, en fonction du nom des sociétés qui les proposent, des mots clés qu'ils contiennent, parfois des sujets auxquels ils se rapportent. *Yahoo*, que nous avons classé dans les moteurs de recherche, fait également office d'annuaire, au même titre que d'autres serveurs plus récents.

L'index complet des sites français se trouve sur l'annuaire de l'Unité REseaux du Cnrs : l'UREC (`http://www.urec.fr`). Il permet de retrouver les sites à partir d'un classement alphabétique, thématique ou géographique.

Plusieurs autres sites français proposent des index comparables. Citons en particulier Lokace (`http://www.iplus.fr/lokace`), Nomade (`http://www.nomade.fr`).

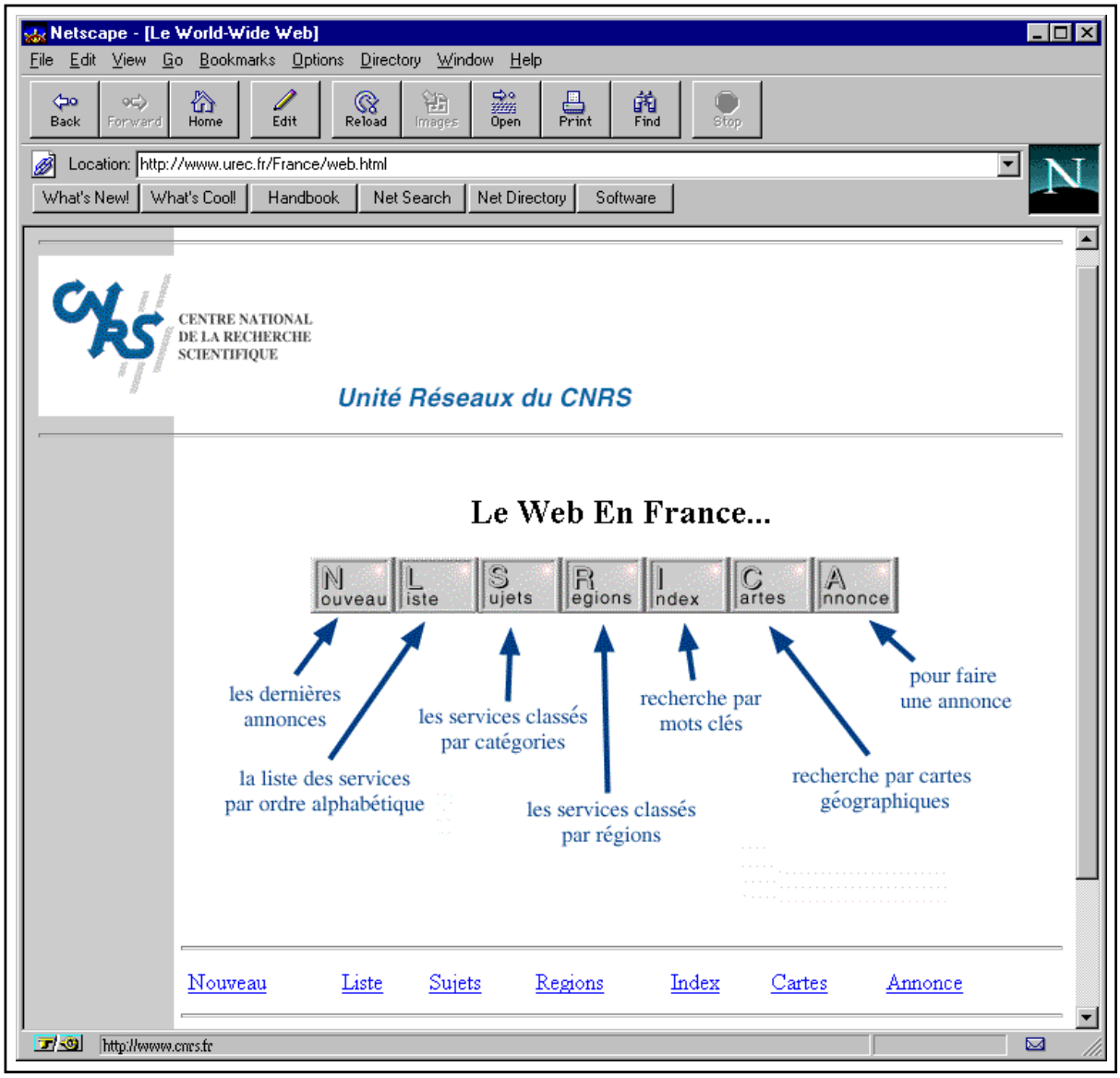

Figure 8.19 *Index de l'Unité Réseaux du CNRS*

Les moteurs de recherche

Il existe deux grands types de moteurs de recherche.

Les premiers archivent automatiquement les pages html (ou plus souvent les premières lignes de ces pages) qu'ils vont récupérer sur le Web. On retrouve alors l'information à partir d'une chaîne de caractères à rechercher dans l'ensemble de ces pages.

Le résultat est souvent indigeste puisque selon l'usage, la fréquence des mots, ces outils peuvent fournir des listes de références impressionnantes, jusqu'à plusieurs centaines de milliers d'entrées. En revanche, lorsqu'on souhaite obtenir un resultat exhaustif, ils sont indispensables.

Altavista

Altavista (`altavista.digital.com`) est un service de Digital Equipment Corporation qui référence environ 30 millions de pages trouvées sur plus de 275 000 serveurs. Une de ses particularités, en dehors de l'extraordinaire puissance de calcul mise à sa disposition qui lui confère une rapidité impressionnante, est d'archiver également plus de quatre millions d'articles provenant de 14 000 forums. On peut considérer que les informations fournies par Altavista sont à jour en permanence (au plus quelques jours de décalage). Bien entendu, l'accès à ce serveur est entièrement gratuit, comme sur la quasi totalité des sites Web de l'Internet.

Créé en février 96, Altavista est aujourd'hui le moteur de recherche le plus visité. Les résultats qu'il fournit sont souvent trop volumineux pour être directement exploitables, mais il offre la possibilité d'affiner une recherche en ajoutant des critères complémentaires. Le paragraphe suivant montre un exemple d'utilisation d'Altavista.

Les seconds utilisent les services d'une équipe de documentalistes qui consultent les pages récupérées par le moteur d'exploration, avant de les classer par thème. Ce classement prend évidemment du temps, ce qui fait que ces serveurs proposent une information un peu moins *fraîche* que les autres moteurs de recherche : il faut compter environ un mois avant qu'un nouveau site soit référencé sur Yahoo.

Yahoo

Créé en 95, *Yahoo* est l'autre référence en matière de moteurs de recherche. Il utilise maintenant la technique développée par Digital pour indexer les documents, et lorsqu'une information ne figure pas dans ses propres bases, il propose les résultats fournis par Altavista.

Son classement thématique le différencie radicalement du serveur de Digital. Là où Altavista fournit une liste de 300 000 références, Yahoo n'en donnera souvent que quelques centaines (certes encore beaucoup trop pour être réellement exploitables...). De plus, il est possible de rechercher un document, non pas par mot clé, mais en descendant dans l'arborescence des catégories, à la manière d'une encyclopédie thématique.

Un moteur de recherche français

La société Écila (`http://www.ecila.com`) propose un moteur de recherche francophone, qui n'indexe que le contenu des sites Français (`http://france.ecila.com`).

9

Les serveurs HTTP

Tout comme avec FTP, les communications sur le Web sont régies par un protocole, nommé HTTP (Hypertext Transfer Protocol), et font appel à une architecture client-serveur fondée sur un système de requêtes et de réponses.

Fondamentalement, un serveur HTTP n'est rien d'autre qu'un serveur de fichiers à l'écoute sur un port TCP (généralement le port système **80**, réservé au protocole HTTP), et qui, lorsqu'il reçoit une demande de connexion suivie d'une requête de la forme :

```
GET /chemin/d/acces/nom.de.fichier
```

répond en expédiant le fichier demandé. À la différence d'une connexion FTP, qui s'établit sur deux ports différents respectivement pour le contrôle et les données, la connexion HTTP est unique – on pourrait dire « semi duplex ». Bien entendu, cette description est très schématique et le protocole HTTP définit de nombreux paramètres et options pour toutes les communications entre le navigateur et le serveur.

Nous n'étudierons qu'une partie de ces spécifications, plus précisément le format général des messages, les valeurs d'état (*status report*), les principales méthodes et les techniques d'authentification.

Nous compléterons cette description par celle de la norme CGI qui permet d'interfacer un serveur HTTP avec des programmes destinés par exemple à consulter des bases de données ou plus généralement à fabriquer des pages HTML à la volée.

Les modèles de serveurs choisis pour illustrer cette section sont Netscape Commerce Server et Apache, tous deux sous Unix. L'installation de Netscape sous Windows NT ne pose pas plus de problèmes, il est donc inutile de la détailler.

Apache est un serveur du domaine public, particulièrement performant et aux fonctions très riches. Au moment où nous rédigeons ces pages, la version NT n'est pas encore disponible, mais elle devrait l'être d'ici la fin 96.

On pourra retenir le serveur Netscape, dont les options par défaut sont généralement convenables, pour éviter de rentrer dans les détails techniques d'une installation. Les paragraphes relatifs à la configuration d'un serveur Apache présentent cependant un grand nombre de notions fondamentales communes aux deux serveurs.

9.1 Le protocole HTTP

Les échanges selon le protocole HTTP sont fondés sur des messages composés d'un en-tête et d'un *corps*, dans un sens comme dans l'autre de la communication (ce qui signifie en particulier qu'un client peut envoyer un document au serveur aussi bien que le recevoir). Le protocole complet est défini pour ses versions 1.0 et 1.1 dans les *drafts* de l'IETF correspondants.

9.1.1 Généralités

Une communication HTTP type entre un client et un serveur se déroule comme suit.

Les deux exemples se rapportent respectivement aux versions 0.9 et 1.0 du protocole ; la différence majeure entre elles est l'utilisation à partir de la norme 1.0 de règles proches de celles du standard MIME pour la forme des messages : on y retrouve les champs *Content-type*, *Content-length*, etc.

On notera cependant que les normes 1.0 et 1.1 du protocole ne respectent pas entièrement le standard MIME (en particulier, il n'y est pas question de).

Comme nous pouvons le constater, la différence essentielle entre les deux types d'échanges réside dans la présence d'un en-tête (a) en HTTP/1.0. Cet en-tête est séparé du corps du message par un saut de ligne (b, d). La requête porte ici sur la page d'accueil du site, plus exactement sur le répertoire racine, donc sur l'index de ce répertoire, qui peut être une page fabriquée par le serveur et indiquant la liste des fichiers, ou un sommaire pour le site (*home page*).

Une requête HTTP 0.9 ne précise pas la version du protocole. Par ailleurs, il est impossible d'y adjoindre aucun champ d'option, puisqu'il n'y aurait pas moyen de séparer un quelconque en-tête du document lui-même. La version 0.9 du protocole HTTP n'est pratiquement plus utilisée, sauf par quelques modèles de navigateurs en mode texte. Nous nous y attarderons d'autant moins qu'il y a peu à en dire de plus que ce que montre l'exemple précédent.

Dans la version 1.0, comme dans la future version 1.1, l'échange est paramétré par des champs de format analogue aux champs d'un en-tête MIME. Dans l'exemple ci-dessus, on

Version 0.9

Client (1)
```
GET /
```
Serveur (2)
```
<html>
Ceci est une page HTML
</html>
```
(fin de communication)

Version 1.0

Client (a)
```
GET / HTTP/1.0
Accept:image/gif, image/x-xbitmap, image/jpeg, */*
User-Agent:Mozilla/2.01Gold (Win95; I)
```
(b) ¶

Serveur (c)
```
HTTP/1.0 200 OK
Date: Thu, 29 Aug 1996 20:12:25 GMT
Server: Apache/1.1.1
Content-type: text/html
¶
```
(d)
```
<html>
Ceci est une page HTML
</html>
```
(fin de communication)

Figure 9.1 *Forme générale d'un échange HTTP*

retrouve le Content-type déjà décrit, avec le type propre au pages du Web : text/html.

9.1.2 En-têtes HTTP

Le seul champ véritablement indispensable dans un en-tête est le `Content-type`. En théorie, il pourrait même ne pas apparaître, mais on conçoit que les clients aient du mal à déterminer le type d'un fichier d'après son contenu : si certains navigateurs sont capables de distinguer un fichier texte d'un fichier html en tentant de détecter les fameuses instructions de mise en page, l'opération devient plus délicate lorsqu'il s'agit de faire la différence entre un source PostScript et un fichier texte ou un graphique HPGL.

La plupart du temps, ce champ est inséré dans l'en-tête par le serveur lui-même (nous verrons lors de l'installation comment préciser au serveur les relations entre extensions de fichiers et types MIME). Cependant, nous verrons aussi que dans le cas de pages HTML fabriquées à la demande par des programmes externes, on peut souhaiter remplir soi-même ce champ. Il faut alors bien sûr respecter la syntaxe des types et s'assurer de ne pas envoyer au client un document qu'il ne pourra pas comprendre.

C'est pour cette raison que les navigateurs envoient au serveur le champ `Accept`. Sous sa forme la plus simple il se présente comme une série de types MIME séparés par des virgules.

Ce champ peut s'étendre sur plusieurs lignes.

Dans l'exemple 9.1 page précédente, Netscape envoyait la série :

```
Accept:image/gif, image/x-xbitmap, image/jpeg, */*
```

Ce qui signifie : « j'accepte n'importe quel type de document, avec une préférence pour les images de type `gif`, `xbitmap` et `jpeg` ». Si le serveur dispose de deux versions des images qu'on lui demande, par exemple `gif` et `tiff`, il peut alors choisir d'envoyer le `gif` afin de répondre au mieux à la demande du navigateur. Notons cependant que peu de serveurs prennent en compte ce champ qui est plutôt destiné aux programmes qui fabriquent des documents à la volée. Ainsi un programme relié au Web et capable de générer à la volée un plan de la circulation à Paris au moment de la requête peut choisir de fabriquer un fichier `gif` plutôt qu'un fichier `tiff`, s'il tient compte de la valeur du champ `Accept`. Nous verrons lorsque nous évoquerons la norme CGI comment récupérer cette valeur au sein d'un script Perl ou d'un programme en C.

Signalons qu'il existe une forme plus complexe du champ `Accept`, qui permet de définir des priorités et des tailles limites. Nous nous contenterons de donner l'exemple suivant, on pourra se reporter aux *drafts* déjà cités pour plus d'informations.

```
Accept: text/plain; q=0.5, text/html,
text/x-dvi; q=0.8; mxb=100000, text/x-c
```

Le signe `;` sépare un type MIME d'un paramètre qui lui est applicable. Le paramètre `q` permet de définir un facteur de qualité, et `mxb` indique une taille limite. Le champ ci-dessus se lit :

« Je préfère les documents de type `text/html` et `text/x-c` (*ce sont ceux dont le facteur de qualité est le plus élevé : par défaut, q=1*). Si ces formats n'existent pas, j'accepte de recevoir une version `dvi` (*DeVice Independent*) du document à condition qu'elle ne dépasse pas 100 Ko. Au pire, m'envoyer une version en texte seul. »

Les autres champs qu'on rencontre fréquemment sont :

`Content-length`, suivi de la taille du document en octets (il s'agit bien entendu de la taille du corps du message, pas de la somme en-tête+document), envoyé généralement par le serveur ;

`Referer` qui émane du client et qui indique un URL : lorsqu'on demande un document en cliquant sur un lien hypertexte, c'est l'URL de la page qui contenait ce lien qui est envoyé ;

`User-agent` qui émane également du client et qui indique son modèle (ainsi pour Netscape Gold sous Windows 95 : User-Agent:Mozilla/2.01Gold (Win95; I)) ;

`Host` qui est échangé par le client et le serveur, et qui sert à indiquer le véritable nom du serveur dans le cas d'un nom virtuel (*Virtual Host*).

Les serveurs proxy utilisent de nombreux champs pour échanger des dates de mise à jour, des durées de validité, etc. (voir la liste précisée dans le *draft* HTTP/1.1)

9.1.3 Méthodes HTTP

À chaque requête émanant du client correspond une action particulière : il peut s'agir de répondre à une demande de fichier (GET), de recevoir un fichier émis par le client (PUT), ou encore d'associer un bloc d'information à une ressource (méthode POST).

La méthode GET indique au serveur qu'il doit renvoyer l'information relative à, ou produite par l'URL indiqué. Si l'URL pointe vers un fichier, c'est le contenu du fichier qui est transmis. S'il pointe vers un programme destiné à fabriquer de l'information, c'est bien entendu l'information produite qui est transmise, non le code source (ou le fichier binaire) du programme.

Ainsi, on peut utiliser la méthode GET pour demander de l'information à un programme chargé de la fabriquer à la volée. Nous reparlerons de cette possibilité lorsque nous étudierons les scripts CGI et le passage d'arguments.

La méthode POST est quant à elle utilisée par les clients pour envoyer de l'information.

Dans ce cas, le message sera constitué de l'en-tête habituel, plus d'un corps contenant l'information proprement dite. Le serveur doit utiliser le corps du message pour, soit le transmettre à un programme, soit éventuellement le conserver dans un fichier.

Dans l'état actuel des choses, cette méthode est généralement utilisée en remplacement de la méthode GET pour transmettre le contenu des champs d'un formulaire à un script CGI.

On rencontre en outre deux autres méthodes, respectivement PUT et HEAD.

La première est utilisée par certains navigateurs pour envoyer un fichier au serveur afin qu'il l'écrive sur disque : l'éditeur HTML de Netscape Gold dispose d'une option `Publish` qui utilise cette méthode pour mettre à jour un document sur un serveur distant (en réalité, l'éditeur laisse le choix entre les protocoles FTP et HTTP). Elle n'est en revanche pas disponible sur de nombreux serveurs.

La seconde est utilisée pour ne demander que l'en-tête d'un message : ainsi, il est possible d'obtenir des informations sur un document sans le télécharger. Les moteurs de recherche et les serveurs proxy exploitent cette possibilité pour détecter si un document a été modifié.

9.1.4 Le statut des requêtes

Comme dans toute communication, une erreur peut parfaitement se produire. À cet effet, et tout comme FTP, le protocole HTTP définit une liste de codes d'erreur renvoyés par le serveur pour indiquer le résultat de la requête. Le serveur y ajoute souvent quelques lignes de code HTML qui sont destinées à l'utilisateur et qui expliquent en clair quelle erreur s'est produite.

```
gide: $ telnet www.fenetre.fr 80
Trying 10.0.0.1...
Connected to www.fenetre.fr.
Escape character is '^]'.
GET /n_existe_pas.html HTTP/1.0
¶
HTTP/1.0 Not found
Date: Thu, 29 Aug 1996 22:03:49 GMT
Server: Apache/1.1.1
Content-type: text/html
¶
<HEAD><TITLE>File Not found</TITLE></HEAD>
<BODY><H1>File Not found</H1>
The requested URL /n_existe_pas.html was not found on this server.<P>
</BODY>
Connection closed by foreign host.
gide: $
```

Figure 9.2 *Erreur HTTP 404*

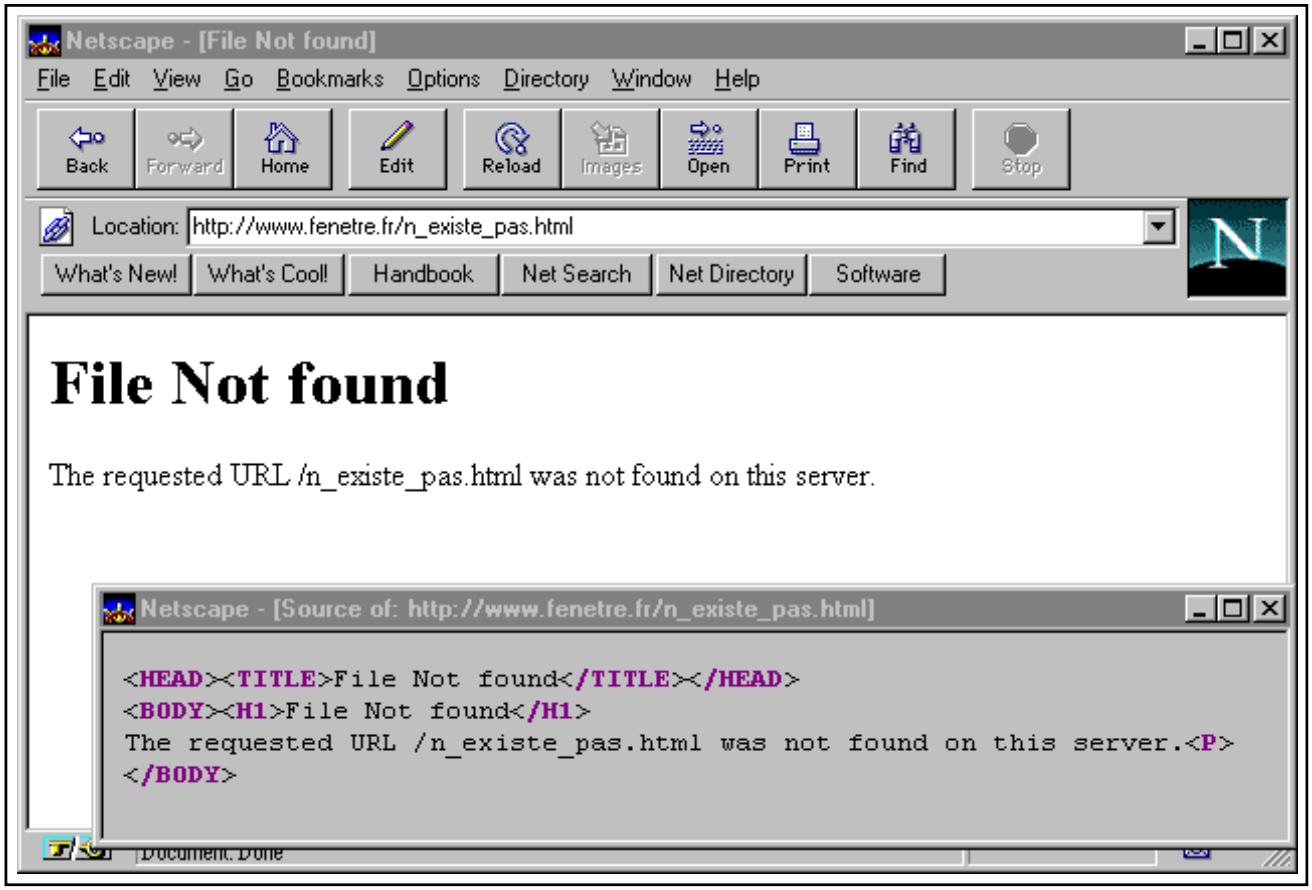

Figure 9.3 *Message d'erreur envoyé par un serveur HTTP (document non trouvé)*

Ces codes suivent la mention HTTP/1.x dans l'en-tête de la réponse du serveur, comme le montre l'exemple précédent. Ils sont répartis dans cinq catégories, de 1xx à 5xx.

Les codes de `100` à `199` désignent de simples messages d'information, par exemple lorsqu'un changement de protocole doit intervenir. On ne les rencontre que très rarement.

Les codes de `200` à `299` indiquent que la requête a été reçue, comprise et acceptée. Le message doit donc contenir la réponse.

```
GET / HTTP/1.0
¶
HTTP/1.0 302 Found
Date: Thu, 29 Aug 1996 22:16:28 GMT
Server: Apache/1.1.1
Location: fr/index.htm
Content-type: text/html
<HEAD><TITLE>Document moved</TITLE></HEAD>
<BODY><H1>Document moved</H1>
The document has moved <A HREF="fr/index.htm">here</A>.<P>
</BODY>
```

Figure 9.4 *Redirection en HTTP (status 302)*

Les codes de `300` à `399` marquent une *redirection*, c'est-à-dire que l'URL demandé n'a pas été trouvé mais que le serveur sait à qui il faut s'adresser pour l'obtenir, ou encore que le document n'a pas changé, donc qu'il n'est pas utile de le récupérer de nouveau. On rencontre parfois le code 302, qui indique qu'un document a été momentanément déplacé. On l'utilise pour renvoyer la demande vers un autre document, voire un autre serveur, en indiquant le nouvel URL dans le champ `Location`.

Sous Netscape, la page HTML envoyée par le serveur pour signaler l'erreur n'apparaîtra pas, car le navigateur ira de lui-même chercher le nouveau document.

Les codes de `400` à `499` indiquent une erreur au niveau du client. Il peut s'agir d'une requête mal formulée, par exemple un URL dont la syntaxe n'est pas correcte ou qui pointe vers un fichier qui n'existe pas, mais également d'un problème d'authentification ou encore d'un problème réseau ayant entraîné une attente trop longue.

Les codes de `500` à `599` se rapportent quant à eux à une erreur au niveau du serveur et signifient généralement un problème d'installation, de configuration ou de stabilité. Il peut cependant s'agir simplement d'une requête dont la syntaxe est légale mais qui fait référence à une méthode que le serveur ne sait pas mettre en œuvre : ainsi le serveur Apache 1.1 refusera la méthode PUT et renverra un code d'erreur `501 (Not Implemented)`.

9.1.5 Les URL dans le protocole HTTP

La notion d'URL est très générale, et chaque protocole définit un champ d'application pour le format décrit à la section 8.2 page 257.

Pour le protocole HTTP, les URL prennent un des aspects suivants :

```
http://machine/
http://machine:port/
http://machine/chemin
http://machine:port/chemin
```

- `machine` désigne un nom de machine ou une adresse IP valide ;
- `port` désigne un numéro de port TCP, le port par défaut étant 80 ;
- `chemin` désigne le chemin d'accès à une ressource (répertoire, fichier, programme...) située sur le serveur.

URL absolus et relatifs

Les différentes références utilisées au sein d'un document HTML désignent généralement des fichiers situés sur le même serveur et dans la même arborescence que la page elle-même. Ainsi les images sont parfois placées dans le même répertoire que les pages, ou dans un sous-répertoire direct. Il n'est alors pas nécessaire de préciser l'URL complet de ces fichiers, puisque la partie commune est déjà déterminée par l'endroit où la page a été récupérée. On parle alors de référence relative, et on ne précise que la portion de l'URL qui diffère de celui de la page.

Ainsi on pourra faire appel au sein d'une page dont l'URL est :

```
http://www.fenetre.fr/pages/page_1.html
```

à l'image `nouveau.gif` située dans le sous-répertoire `im/` du répertoire `pages` sur le serveur Web de la société FeNETre en indiquant simplement le paramètre HTML convenable, `HREF="im/nouveau.gif"`.

9.1.6 Buts et principes de l'authentification

L'authentification est une technique utilisée pour restreindre l'accès à certaines ressources ou documents grâce à des codes d'accès et des mots de passe. Le protocole HTTP définit des champs spécifiques pour les en-têtes de message afin d'échanger l'information relative à l'authentification de l'utilisateur.

Concrètement, tout se passe comme si on devait entrer son code d'accès (ou *login*) et son mot de passe pour se connecter à un système Unix.

Si le couple code d'accès + mot de passe n'est pas correct, on n'a tout simplement pas accès à l'URL concerné (notons que généralement, on protège des répertoires complets). Le

Figure 9.5 *Authentification par code d'accès et mot de passe*

serveur renvoie alors un code (`401 : Unauthorized`) et un message d'erreur. Le navigateur peut proposer à l'utilisateur de s'identifier à nouveau (figure 9.6) ou afficher le message (figure 9.7).

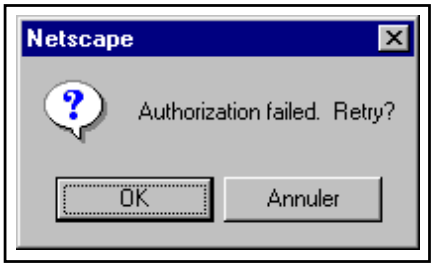

Figure 9.6 *Échec de l'authentification HTTP*

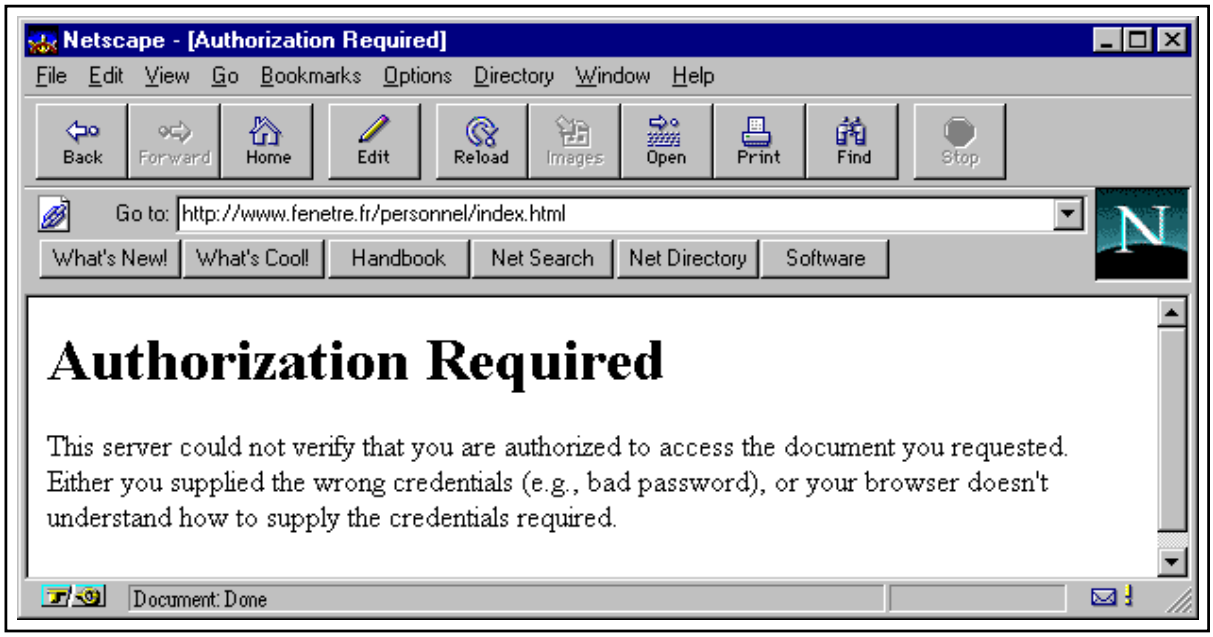

Figure 9.7 *Message d'identification incorrecte généré par un serveur HTTP*

Voilà pour ce qui concerne la partie apparente. Cependant au niveau de l'échange d'informations entre le navigateur et le serveur, les choses sont un peu plus complexes : il est parfois délicat de faire transiter le mot de passe *en clair* sur le réseau – il serait bien trop simple de l'intercepter pour quelqu'un ayant accès au système : par exemple en installant un « faux proxy », ou encore en surveillant les paquets TCP, ou par un nombre incalculable d'autres techniques. Plusieurs méthodes sont alors disponibles, depuis le simple cryptage de la phase d'authentification jusqu'au cryptage complet de la communication par un système de clés publiques et privées. Nous ne verrons que le système le plus élémentaire, sans cryptage, supporté par presque tous les serveurs.

Basic authentication

La technique d'authentification la plus simple pour le protocole HTTP consiste à n'échanger que des mots de passe "en clair", encodés afin de garantir leur intégrité avec un procédé similaire au codage `Base64`. Bien entendu, elle n'offre qu'un niveau de sécurité très relatif. En effet, rien n'empêche un utilisateur malveillant de surveiller les communications pour récupérer le mot de passe, la méthode de décodage étant simple et bien connue. Son principal avantage est sa simplicité d'emploi au niveau du serveur, puisqu'il suffit à réception du mot de passe de vérifier la validité de ce dernier dans une base de données qui se résume le plus souvent à un simple fichier texte semblable au fichier `/etc/passwd` d'Unix. Cette technique est généralement nommée *Basic authentication*. Dans Apache, elle correspond au module `mod_auth`, qui sauf mention contraire est compilé avec le reste du serveur. La figure 9.8 montre un exemple de phase d'authentification en HTTP.

```
gide: $ telnet www.fenetre.fr 80
Trying 10.0.0.1...
Connected to www.fenetre.fr.
Escape character is '^]'.
GET /personnel/index.html HTTP/1.0
Authorization:Basic d3d3OmhhcnJhcHM=
¶
HTTP/1.0 200 OK
Date: Thu, 29 Aug 1996 23:39:44 GMT
Server: Apache/1.1.1
Content-type: text/html
Content-length: 1035
Last-modified: Thu, 29 Aug 1996 22:45:45 GMT
¶
<html>
<head>
<title>FeNETre - Bienvenue</title>
</head>
[...]
```

Figure 9.8 *Echange HTTP avec authentification*

Ce sont les champs `WWW-Authenticate` et `Authorization` qui permettent la *négociation* du droit d'accès. La première connexion au serveur est une requête simple. Le serveur

répond par une erreur `401 (Unauthorized)` pour signifier que l'accès est réservé, en ajoutant la mention `WWW-Authenticate` à l'en-tête du message afin de préciser qu'il faut procéder à l'authentification de l'utilisateur, et surtout afin de fournir le label (*realm*) auquel se rapporte cette authentification. Le navigateur affiche alors la boîte de dialogue code d'accès + mot de passe afin de demander ces informations à l'utilisateur, puis il formule de nouveau sa requête en indiquant le mot de passe encodé dans le champ `Authorization`. Le processus se répète tant que le mot de passe n'est pas correct et que l'utilisateur redemande le document.

Authentification incorrecte

```
GET /personnel/index.html HTTP/1.0
¶
HTTP/1.0 401 Unauthorized
WWW-Authenticate: Basic realm="serveur interne de la société FeNETre"
```

Authentification correcte

```
GET /personnel/index.html HTTP/1.0
Authorization:Basic d3d3OmhhcnJhcHM=
¶
HTTP/1.0 200 OK
```

Figure 9.9 *Séquences d'authentification HTTP*

Notons que le champ `WWW-Authenticate` précise deux choses : tout d'abord la méthode utilisée (ici *Basic*), ensuite le label (*serveur interne de la société FeNETre*). Le label indique à quelle partie protégée du serveur on souhaite accéder. Il permet de découper l'ensemble des documents dont l'accès est restreint en sous ensembles, accessibles à des catégories d'utilisateurs différentes, avec des codes d'accès et des mots de passe différents, voire des méthodes d'authentification différentes. Selon la méthode spécifiée, le navigateur devra décider d'envoyer le mot de passe encodé, ou bien de procéder à une authentification par clé publique, cryptage MD5, RSA, etc.

9.2 Installation et configuration d'un serveur HTTP

Les paragraphes suivants passent en revue les différentes options de configuration et d'installation de deux serveurs, respectivement Netscape Commerce Server et Apache 1.1 (sous Unix).

Le serveur Netscape présente une installation simple et conviviale pilotée par l'intermédiaire d'un navigateur Web. Sa mise en œuvre ne devrait poser aucun problème particulier, qu'il s'agisse d'un système Unix ou de Windows NT.

Le serveur Apache nécessite quant à lui plus de précautions. En revanche, il est totalement gratuit, et bénéficie d'une richesse d'options impressionnante et en constante évolution. C'est de surcroît l'un des serveurs les plus performants disponibles à l'heure actuelle. À l'instar de l'API de Netscape, qui permet d'ajouter au serveur des modules externes, le serveur Apache

peut être complété par des modules intégrés au fichier exécutable, ou chargés dynamiquement lors de l'utilisation.

Un autre serveur particulièrement connu est le serveur NCSA. C'est de lui qu'émane à l'origine le serveur Apache, et les deux sont restés assez proches quant au format de leurs fichiers de configuration. Son installation ne devrait donc pas poser plus de problèmes.

9.2.1 Les principaux répertoires, paramètres et fichiers

Tous les serveurs HTTP ont en commun un certain nombre de directives, qui peuvent porter des noms légèrement différents dans les fichiers de configuration. Il s'agit principalement des répertoires dans lesquels sont conservés les fichiers de configuration, les relevés de connexion et les pages du serveur, ainsi que quelques options relatives au protocole lui-même.

Server Root

Ce répertoire indique le chemin d'accès complet où se trouvent généralement l'exécutable, les fichiers de configuration, et les relevés de connexion (fichiers de *log*) du serveur. Par défaut, les serveurs du CERN, NCSA et Apache font référence au répertoire :

```
/usr/local/etc/httpd
```

dont les sous-répertoires `conf` ou `config` contiennent respectivement les fichiers de configuration et les scripts.

Document Root

Ce répertoire est celui sous lequel sont conservés les documents du serveur : pages HTML, images, vidéos... Comme nous le verrons, sa valeur est disponible dans une variable d'environnement lors de l'appel d'un programme externe.

C'est à ce répertoire que se réfère ce dont nous désignerons par l'expression « racine du serveur », c'est-à-dire le chemin d'accès précisé au sein des URL.

Ainsi, `http://www.fenetre.fr/` représente le répertoire `/usr/local/www_docs/` du serveur de la société FeNETre, et :

```
http://www.fenetre.fr/personnel/salaires/grille.html
```

indique le fichier :

```
/usr/local/www_docs/personnel/salaires/grilles.html
```

Par défaut, les serveurs du CERN, NCSA et Apache font référence au répertoire :

```
/usr/local/etc/httpd/htdocs
```

Script Alias

Cette option définit une correspondance entre un répertoire particulier relatif à la racine du serveur et généralement appelé `cgi-bin`, et un répertoire sur le disque de la machine. Ce répertoire a un statut spécial, puisqu'il est destiné à héberger des programmes ou scripts CGI utilisés par exemple pour traiter des formulaires, accéder à des bases de données...

Sa raison d'être est simple : il s'agit de séparer les documents des programmes, pour des raisons classiques de sécurité et de droits d'accès. Puisque certains programmes CGI peuvent présenter des défaillances au niveau de la sécurité, il peut être nécessaire de leur conférer des permissions spéciales, ou de les attribuer à un utilisateur et un groupe *inoffensif*.

De plus, il peut s'avérer préférable de distinguer les concepteurs du site Web autorisés à mettre en place des scripts de ceux qui ne doivent pas pouvoir le faire (pour des raisons de compétence ou de fonction).

Tous les fichiers figurant dans ce répertoire sont considérés comme des programmes – ce qui ne dispense pas, bien sûr, de leur affecter les droits en exécution appropriés, que ce soit sous Unix ou sous Windows NT. Ils peuvent porter n'importe quelle extension (y compris `html`, bien que cela soit fortement déconseillé...).

La quasi totalité des serveurs autorise maintenant l'utilisation de scripts ou programmes CGI en dehors des répertoires dédiés à cet usage. C'est à l'administrateur du site de décider si cette option ne présente pas de risque.

Notons que dans le cas de scripts CGI placés dans les mêmes répertoires que les pages, il est nécessaire de définir des extensions particulières sans lesquelles l'exécutable ne sera pas lancé. Cela afin d'éviter que de simples pages HTML auxquelles on aurait par erreur donné les droits en exécution ne soient appelées comme des programmes.

L'extension la plus couramment choisie est `.cgi`.

Server Name

Il est fréquent d'attribuer à une même machine à la fois le rôle de serveur pour le Web, de passerelle *email*, voire de serveur de fichiers, etc. Dans un tel cas, la machine en question portera vraisemblablement plusieurs noms.

Ainsi, chez FeNETre, la machine `sundi.fenetre.fr` sert de serveurs Web et FTP, de DNS et de MX pour la zone `fenetre.fr`.

Elle porte donc également les noms CNAME suivants :

```
www.fenetre.fr
ftp.fenetre.fr
ns.fenetre.fr
relay1.fenetre.fr
```

Lorsqu'au démarrage le serveur HTTP cherche à connaître le nom de la machine sur laquelle il se trouve, il interroge le système qui lui renvoie `sundi.fenetre.fr`.

La directive `ServerName` permet d'imposer le nom du serveur, elle sert en particulier pour mettre en place plusieurs sites sur un même serveur avec des noms de domaines différents (notion de *Virtual Host*). Notez cependant que le nom choisi doit **impérativement** correspondre à une entrée valide dans le DNS. N'oubliez donc pas d'ajouter un CNAME de ce nom vers le nom de la machine.

Les fichiers de log

À l'instar des télécopieurs ou des standards téléphoniques, les serveurs HTTP maintiennent des registres d'activité et des relevés de connexions qui recensent toutes les requêtes reçues par le système, qu'elles aient été servies ou pas.

Ces fichiers, appelés *logs* du serveur, recensent différents types d'informations, les principales étant les connexions servies et les erreurs de fonctionnement.

Access Log

Le fichier `access_log` contient la liste des requêtes reçues. Il se présente généralement sous un de ces trois formats, respectivement hérités des serveurs NCSA et des serveurs proxy Harvest : Common Log Format ou CLF, Extended Common Log Format et format Harvest (du nom d'un serveur proxy-cache assez répandu).

Error Log

Ce fichier contient la liste des erreurs de fonctionnement décelées par le serveur. Il peut s'agir d'erreurs *normales* ou anodines, telle une requête concernant un document qui n'existe pas ou dont les droits d'accès sont incorrects, ou plus grave, lorsqu'un script fonctionne mal ou même lorsque le serveur lui-même contient un défaut qui génère une anomalie système pouvant aller jusqu'à l'interruption du processus.

Le fichier `error_log` (figure 9.11 page ci-contre) est un outil indispensable en cas de fonctionnement anormal du serveur. Dans certains cas, il pourra même recevoir des informations de *débogage*.

```
Common Log Format

youpi.hopla.fr - - [06/Oct/1995:13:51:23 -0500] "GET /vitrage\
s.html" 200 3296

Common Log Format (Extended)

youpi.hopla.fr - - [19/Sep/1995:15:19:07 -0500] "GET /vitrage\
s.html HTTP/1.0" 200 1752 "" "NCSA\_Mosaic/2.7b1 (X11;IRIX 5.\
3 IP22)  libwww/2.12 modified"

Harvest

host:youpi.hopla.fr;user:-;time:[19/Sep/1995:15:19:07 -0500];\
request:GET /vitrages.html HTTP/1.0;agent: NCSA\_Mosaic/2.7b1\
 (X11;IRIX 5.3 IP22)  libwww/2.12 modified;referer:http://wor\
ldwide.windows.net/everything/France/list.html;status:200;byt\
es:1752
```

Figure 9.10 *Les différents formats de relevés de connexions*

```
[Fri Aug 30 13:46:16 1996] access to /usr/local/www\_docs/men\
uiserie.html failed for youpi.hopla.fr, reason: File does not\
 exist
[Fri Aug 30 15:25:08 1996] request lost connection to client \
youpi.hopla.fr
[Fri Aug 30 15:28:23 1996] access to /usr/local/www\_docs/for\
um/reponse.cgi failed for youpi.hopla.fr, reason: file permis\
sions deny server access
```

Figure 9.11 *Extrait de relevé d'erreurs d'Apache (error_log)*

Chaque fichier enregistre une entrée par ligne, accompagnée de la date et l'heure courante. Dans le fichier `access_log`, date et heure sont au « format Internet », c'est-à-dire :

```
[jj/MM/aaaa:hh:mm:ss +/-tttt]
```

où `tttt` représente la valeur absolue du décalage horaire par rapport à l'heure universelle (les deux premiers chiffres indiquent le nombre d'heures).

9.2.2 Configuration des types MIME

Tout comme pour les clients, il faut préciser au serveur comment retrouver le type MIME d'un document à partir de son extension. C'est le rôle d'un fichier généralement nommé `mime.types` et situé dans le répertoire de configuration du serveur. Il ne contient pas de notion d'action, mais simplement la correspondance type MIME / extension(s).

Le fichier fourni par défaut avec le serveur devrait suffire pour la grande majorité de vos besoins. Si toutefois il était nécessaire d'insérer un format particulier dans la liste, il suffirait d'éditer le fichier et d'ajouter la ligne correspondante, sous la forme :

```
type_mime/sous_type_mime ext1 [ext2 ext3 ...]
```

```
application/postscript          ai eps ps
application/rtf                 rtf
multipart/mixed
text/html                       html htm

[...]
```

Figure 9.12 *Extrait du fichier mime.types*

en séparant les entrées par un ou plusieurs espaces ou tabulations.

9.2.3 Permissions et droits d'accès

La plupart du temps, le serveur HTTP est lancé en tant qu'utilisateur *root*. Les serveurs tels qu'Apache laissent la possibilité d'indiquer *sous* quels identificateurs d'utilisateur (*userid*) et de groupe (*groupid*) ils doivent fonctionner après le lancement.

On considère qu'il est particulièrement dangereux de laisser un serveur fonctionner en tant qu'utilisateur *root*. Les programmes CGI externes s'exécutent en effet avec le même *userid* que le serveur, ce qui constitue une faille importante dans la sécurité générale du système, d'autant que la norme CGI n'est pas sans présenter quelques défauts à ce sujet.

La meilleure approche consiste à demander au serveur de se rabattre après lancement sur les identificateurs *nobody* et *nogroup*, qui désignent généralement un utilisateur dont les possibilités d'accès sont minimales. Bien entendu, il peut y avoir des exceptions, par exemple lorsqu'un programme CGI doit consulter et mettre à jour une base de données appartenant à un autre utilisateur et qu'il n'est pas question de laisser en accès libre. Dans ce cas, on pourra donner les droits particuliers du possesseur de la base, non pas au serveur, mais au programme lui-même (en utilisant par exemple le *sticky-bit* s'il s'agit d'un système Unix), ou encore choisir de faire tourner le serveur au sein d'un groupe créé pour l'occasion et dont l'accès se limite aux fichiers indispensables.

9.2.4 Mode de fonctionnement

À chaque requête correspond un processus serveur (éventuellement un *thread* si l'architecture supporte ce type de processus *allégé*). Bien qu'il soit techniquement possible de construire un serveur capable de répondre à toutes les requêtes à partir d'un seul processus, cette solution est rarement retenue pour de nombreuses raisons : tout d'abord, les requêtes seraient servies l'une après l'autre, ce qui générerait parfois des temps d'attente insupportables pour les utilisateurs, ensuite le serveur serait très sensible aux anomalies de fonctionnement puisqu'une interruption due à une des connexions se répercuterait inévitablement à toutes les autres, et enfin le poids du processus par rapport aux autres démons ou programmes tournant sur la machine serait faible, ce qui limiterait les performances générales du serveur.

Un serveur HTTP peut fonctionner avec `inetd`, ou en mode `standalone`. La différence se situe principalement dans le fait qu'en mode `inetd`, un nouveau serveur sera lancé à chaque requête. Ce mode est acceptable si le site est de taille raisonnable avec une fréquentation moyenne, mais il peut représenter une charge machine trop importante dès lors que le nombre de connexions augmente. En mode `standalone`, la plupart des serveurs gèrent au mieux le nombre de processus qu'ils lancent ; ainsi Apache démarre d'emblée un nombre minimal de processus afin de pouvoir répondre à quelques requêtes sans devoir démarrer un nouveau serveur à chaque fois, puis augmente ce nombre lorsque tous les serveurs sont occupés. Afin de ne pas trop occuper la mémoire avec des processus inactifs en dehors des heures de grande fréquentation, il en supprime automatiquement une partie.

Mode inetd

Le serveur est lancé par le démon `inetd` dès qu'une connexion arrive sur le port concerné. Le serveur communique alors avec l'extérieur par l'intermédiaire de l'entrée et de la sortie standard. Les premiers serveurs fonctionnaient de cette manière, car elle dispense de doter le programme de l'interface chargée de gérer la partie réellement *serveur*, c'est-à-dire de l'interface avec le réseau.

Nous pouvons très simplement fabriquer un serveur HTTP élémentaire grâce à `inetd`. Il suffit de rédiger un petit programme ou script qui lit l'entrée standard, recherche la requête GET et envoie le document demandé. Voici un exemple de script *shell* qui joue ce rôle, en transformant les requêtes sur des répertoires en requêtes sur le fichier `index.html`. On notera qu'il n'est pas question de `Content-type` dans la réponse, en fait il n'y a même pas d'échange d'en-tête. C'est la version 0.9 du protocole HTTP qui est utilisée ici, pour des raisons de simplicité.

```
luc: $ cat serveur.sh

\#!/bin/sh

cd /usr/local/www\_docs
file=‘gawk ’/GET/ \{split(\$0,a," |/|\\r"); print a[3]; exit;
if (test X\$file = X)
       then file="index.html"
fi

if !(/bin/cat ‘echo \$file‘ 2\>\&1)
       then echo Fichier introuvable
fi
```

Figure 9.13 *Mini-serveur HTTP en script shell*

On pourra installer ce programme en éditant le fichier de configuration du démon `inetd` et en y ajoutant une ligne pour le port 80 (service `www`), puis en demandant au démon de mettre ses tables à jour en lui envoyant un signal SIGHUP. Sous Linux, cela donne à peu de choses près le contenu de la figure 9.14 page suivante.

```
luc: $ su
Password: ********
luc: # vi /etc/inetd.conf

[...]

www     stream  tcp     nowait  root    /home/luc/serveur.sh        httpd

[...]

luc: # ps x | grep inetd

 40  ?  S     0:06 /usr/sbin/inetd

luc: # kill -HUP 40
```

Figure 9.14 *Installation du mini-serveur HTTP*

Le serveur devrait maintenant fonctionner ; il est possible de le vérifier avec un *telnet* sur le port 80, ou directement en se connectant avec un navigateur tel que Netscape. La plupart des navigateurs même les plus récents sont capables de communiquer avec la version 0.9 du protocole.

Figure 9.15 *Connexion au serveur de test : aucun fichier HTML n'a été mis en place*

Comme le montre la figure 9.15, nous n'avons pas mis de fichier dans le répertoire du site ! Nous allons donc ajouter un document très simple sous le nom `index.html` dans le répertoire choisi en tant que `Document Root` (ici `/usr/local/www_docs`). Nous verrons plus loin dans ce chapitre comment rédiger des pages HTML, en attendant nous utiliserons le modèle de la figure 9.16 page suivante.

Une fois cette page ajoutée, notre mini-serveur fonctionne correctement, comme le montre la figure 9.17 page ci-contre.

```
luc: $ cat /usr/local/www_docs/index.html

<HTML>

<HEAD>
<TITLE>Page d'accueil</TITLE>
</HEAD>

<BODY>
<H1>
Bienvenue à la société fenêtre!
</H1>
Bientôt ici même toutes les informations sur notre société
</BODY>

</HTML>
```

Figure 9.16 *Prototype de page HTML*

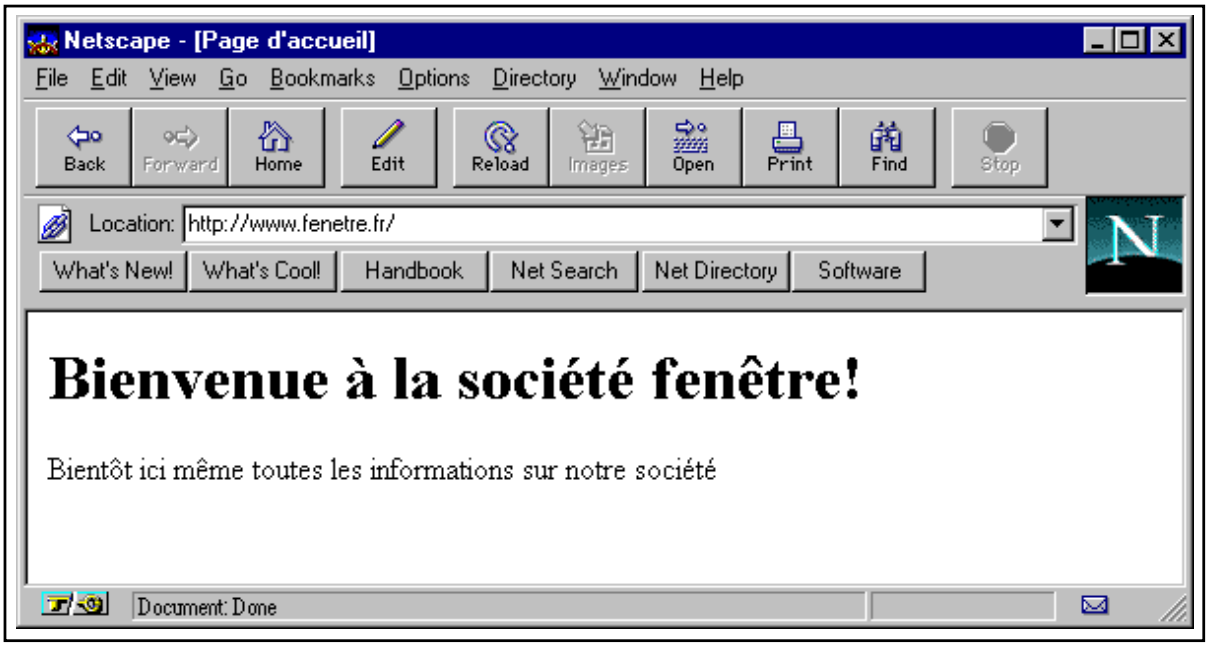

Figure 9.17 *Document envoyé par le mini-serveur de test*

Mode standalone

Dans ce mode le serveur est lancé une fois pour toutes (par exemple au démarrage de la machine, par l'intermédiaire des fichiers de configuration du système).

Il reste à l'écoute des requêtes qui pourraient arriver sur le port 80 (ou tout autre port configuré spécialement) et, lorsqu'il en reçoit une, se duplique en lançant un second processus (*fork*) afin d'y répondre.

Nous l'avons déjà mentionné : les serveurs récents utilisent une technique dite de *pre-forking* qui permet de ne pas dupliquer le processus au moment où la requête arrive. La réponse est ainsi envoyée plus rapidement, et le système est moins sollicité.

9.3 Installation et configuration d'Apache

Il faut d'abord télécharger le serveur Apache version 1.1.1 en FTP anonyme depuis, par exemple :

```
ftp://ftp.ibp.fr/pub/www/apache/dist/apache_1.1.1.tar.gz
```

Une distribution binaire est disponible pour de nombreux systèmes ; cependant, les paragraphes suivants montrent une installation complète à partir des sources, indispensables pour intégrer au serveur – et déboguer le cas échéant – des modules complémentaires. Pour compiler et installer le serveur, il faudra bien entendu un compilateur C, par exemple `gcc` de GNU. Il est préférable d'utiliser un compilateur réellement compatible avec le standard ANSI.

Dans un premier temps, le serveur est décompressé et désarchivé dans un répertoire personnel.

```
luc: $ gzip -dc apache_1.1.1.tar.gz | tar xvf -
apache_1.1.1/
apache_1.1.1/cgi-bin/
apache_1.1.1/cgi-bin/printenv
apache_1.1.1/cgi-bin/test-cgi
apache_1.1.1/conf/
apache_1.1.1/conf/access.conf-dist
apache_1.1.1/conf/httpd.conf-dist
apache_1.1.1/conf/mime.types
luc: $ cd apache_1.1.1/src
luc: /apache_1.1.1/src$
```

Figure 9.18 *Installation des sources d'Apache*

Compilation et test

La figure 9.18 montre le désarchivage des sources de l'application. Après cette étape, il faut éditer le fichier `Configuration` en suivant les commentaires.

Si on n'utilise pas le compilateur de GNU, il faudra préciser le nom de commande du compilateur retenu. On n'oubliera pas d'indiquer pour quelle plate-forme le programme doit être compilé, en retirant les # devant les options appropriées. Ainsi avec gcc pour Linux le fichier prend l'aspect de la figure 9.19 page ci-contre (nous avons nous-mêmes ajouté l'option `-m486` afin d'optimiser le code pour les processeurs 486 ou Pentium).

La fin du fichier permet d'indiquer quels modules ajouter au serveur. Ces modules font partie intégrante du serveur Apache après compilation. Cette notion correspond simplement à des fichiers C relativement indépendants du reste du serveur, et qui utilisent des fonctions dont les prototypes sont standardisés. Elle permet d'ajouter des fonctionnalités au serveur sans devoir toucher au *noyau* du code.

Nous conserverons la liste des modules telle quelle pour le moment, et nous verrons par la suite comment gérer l'ajout d'un nouveau module ou le remplacement d'un module existant.

Il faut maintenant lancer la commande `Configure` (figure 9.20, qui va utiliser le fichier `Configuration` pour fabriquer le fichier `Makefile`. Le serveur peut alors être compilé avec `make`.

Une fois l'exécutable généré, il faut compléter les fichiers de configuration.

```
[...]

# For Sequent
#AUX_CFLAGS= -DSEQUENT
# For Linux -m486 ONLY IF YOU HAVE 486 BINARY SUPPORT IN KERNEL
AUX_CFLAGS= -DLINUX -m486
# For A/UX
#AUX_CFLAGS= -DAUX -D_POSIX_SOURCE
#AUX_LIBS= -lposix -lbsd -s
# For SCO ODT 3

[...]
```

Figure 9.19 *Extrait du fichier Configuration (serveur Apache)*

```
luc: /apache_1.1.1/src$ Configure
Using 'Configuration' as config file
luc: /apache_1.1.1/src$ make
```

Figure 9.20 *Compilation du serveur Apache*

Configuration

Les fichiers de configuration se trouvent dans le répertoire `conf` de la distribution. Ils sont au nombre de trois, plus le fichier `mime.types` que nous n'avons pas besoin de modifier.

Une première étape consiste à copier les fichiers par défaut à leur emplacement final. Chacun d'eux porte une extension `-dist`, il s'agit simplement de les copier sous le même nom mais sans cette extension :

```
luc: /apache_1.1.1/conf$ cp httpd.conf-dist httpd.conf
luc: /apache_1.1.1/conf$ cp srm.conf-dist srm.conf
luc: /apache_1.1.1/conf$ cp access.conf-dist access.conf
```

Figure 9.21 *Mise en place des fichiers de configuration*

Nous commencerons par le fichier `httpd.conf`. Il s'agit du fichier de configuration principal. C'est lui qui contient les directives concernant la localisation physique du serveur, le numéro de port, etc. Le fichier fourni avec la distribution indique une liste de valeurs par défaut et est assez bien commenté.

Le fichier httpd.conf

Toutes les configurations proposées par la suite supposent que le serveur tourne en mode `StandAlone` et sur le port 80. Certaines directives ont déjà été présentées, la liste ci-dessous ne fait que fournir quelques indications complémentaires sur les options possibles.

– `HostnameLookups`

  Si sa valeur est *on*, cette option indique à Apache qu'il doit rechercher les noms des machines qui lui soumettent des requêtes. Par défaut, le serveur ne dispose pas de ces noms mais uniquement des adresses IP. Il peut être utile de les obtenir, par exemple pour connaître les domaines depuis lesquels on vient voir votre serveur, mais si la connexion à l'Internet est déjà chargée, il est préférable de rendre cette fonction inopérante, car elle est parfois gourmande en ressources (voir le chapitre 4 page 145 sur le DNS).

– `User/Group`

  Ce sont les directives évoquées au paragraphe 9.2.3 page 292. On peut préciser pour chacune un nom d'utilisateur ou de groupe, ou directement le numéro (*uid/gid*) précédé par le signe #.

– `ServerRoot`

  On indique ici le répertoire dans lequel a été installée la distribution d'Apache. Il doit s'agir d'un chemin complet, sans abréviation du genre *user*.

– `PidFile`

  Afin de répondre efficacement à plusieurs requêtes simultanées, la plupart des serveurs HTTP lancent automatiquement plusieurs processus. Au démarrage, Apache ira écrire dans ce fichier le numéro du processus principal. Pour interrompre le serveur, il suffit d'interrompre ce processus, tous les autres suivront.

– `ServerName`

  On n'oubliera pas d'indiquer le nom officiel du serveur s'il est différent du nom de la machine qui l'héberge.

Les différents *timeouts* et nombres min/max de serveurs peuvent conserver leurs valeurs par défaut. Si le site connaît un franc succès, il faudra augmenter les maxima.

Le fichier srm.conf

C'est dans ce fichier que sont précisées les options de configuration relatives à l'organisation du site, qu'il s'agisse du ou des répertoires contenant les documents, ou des types de fichiers qu'ils représentent. La liste des options qu'il est indispensable de mettre à jour est assez courte, puisque la plupart des valeurs par défaut sont celles adoptées par le plus grand nombre de sites. Il faut cependant définir les directives suivantes :

`DocumentRoot`

Le répertoire dans lequel seront placés les documents du serveur. La valeur par défaut est assez répandue mais peu commode à cause de sa longueur. On pourra lui préférer un chemin

d'accès plus court tel que `/home/www`, `/usr/local/www`, `/usr/local/htdocs`, `/usr/local/www_docs`, etc. On pourra aussi établir un lien symbolique de ces chemins vers le répertoire par défaut `/usr/local/etc/httpd/htdocs`.

`UserDir`

Il n'est pas nécessaire de changer la valeur de cette option, on retiendra qu'elle représente le répertoire sous lequel les utilisateurs du système peuvent placer leurs documents (dans le chemin d'accès de leur compte personnel) pour qu'ils soient accessibles sur le Web. La valeur par défaut, `public_html`, est presque universelle. On accède aux fichiers d'un utilisateur en indiquant son identificateur (*login*) précédé du signe , tout comme en *shell* Unix (à la différence près qu'il est impossible de *descendre* du répertoire `public_html` vers la racine du compte de l'utilisateur). Ainsi l'utilisateur `rousseau` pourra placer un fichier `qui_suis_je.html` dans son répertoire `/home/rousseau/public_html` et y accéder par l'URL `http://www.fenetre.fr/ rousseau`.

`ScriptAlias`

Il peut être préférable de centraliser les scripts dans un répertoire spécial plutôt que de permettre aux concepteurs du site de les mélanger à leurs pages. On indiquera par cette directive le nom *symbolique* sous lequel on fera référence à ce répertoire au sein des pages, et le chemin d'accès complet. La valeur par défaut pour l'alias, `/cgi-bin`, est assez répandue. La directive suivante :

```
ScriptAlias /scripts/ /usr/local/web/programmes/
```

signifie que lorsqu'une requête fait référence à l'URL :

```
http://www.fenetre.fr/cgi-bin/execute.sh
```

il s'agit en fait du script `execute.sh` qui se trouve dans le répertoire :

```
/usr/local/web/programmes/execute.sh
```

Attention : il est évidemment possible de faire pointer le répertoire *alias* pour les scripts vers un répertoire qui ne se trouve pas dans l'arborescence du site. Cela ne signifie pas pour autant que s'il se trouve tout de même dans l'arborescence, il ne soit pas utile de l'indiquer. En effet, la directive `ScriptAlias` a pour double objectif de déclarer l'alias, *et* de préciser que le répertoire en question contient des scripts ou programmes (et pas autre chose).

Si en revanche on souhaite qu'il soit possible d'intégrer des scripts aux répertoires contenant les pages, il faut définir la ou les extensions à donner aux fichiers pour qu'ils soient considérés comme des programmes par le serveur. On retirera pour cela le # qui met en commentaire la ligne suivante :

```
AddHandler cgi-script .cgi
```

Notons qu'il est tout à fait possible de donner une autre extension. On pourrait par exemple indiquer à la suite :

```
AddHandler cgi-script .pl
AddHandler cgi-script .sh
```

pour s'assurer que les scripts rédigés respectivement en Perl et en *shell* puissent être ajoutés au serveur tout en conservant leur extension « naturelle ». On retiendra cependant qu'il est préférable de limiter le nombre des extensions de programmes. Si changer l'extension usuelle d'un programme Perl ou d'un script *shell* en `.cgi` peut paraître rébarbatif, il s'agit pourtant d'un moyen efficace de limiter les erreurs et les problèmes de sécurité liés à des programmes qui se trouvent au mauvais endroit.

La notion de `Handler` que nous venons d'évoquer implicitement est propre à Apache. Le concept en est simple, puisqu'il s'agit d'associer une action spécifique à un type de fichier particulier. Ainsi dans le cas des scripts CGI, les fichiers qui portent l'extension adéquate sont *transmis* à un gestionnaire (*handler*) qui se charge de les exécuter.

Un autre gestionnaire fort utile permet de gérer très simplement les cartes cliquables ou Image Maps. Le format et la fabrication de ces cartes sera décrit plus loin, mais signalons dès à présent qu'elles sont contenues dans des fichiers qui doivent habituellement être interprétés par un programme externe. Les serveurs récents (depuis la version 1.5 du NCSA) intègrent ce programme. Le gestionnaire `imap-file` d'Apache agit sur les fichiers qui décrivent les cartes et qui portent généralement l'extension `.map`. Nous utiliserons cette fonctionnalité dans un exemple, nous pouvons donc supprimer le # qui met en commentaire la ligne suivante :

```
AddHandler imap-file map
```

Le fichier access.conf

Ce fichier règle les questions de sécurité d'accès. Il permet d'indiquer quels répertoires sont accessibles à tous les utilisateurs, de définir des restrictions pour d'autres répertoires plus sensibles, voire d'en interdire complètement l'accès à toutes les machines qui ne feraient pas partie du domaine (au sens Internet) de la société (fonction qui se rapproche d'une utilisation en intranet).

Les directives de ce fichier s'articulent au sein de blocs qui précisent leur portée. Ainsi, toutes les directives incluses dans le bloc :

```
<Directory /usr/local/www_docs/personnel>

[...]

</Directory>
```

(où la notation `</Directory>` empruntée au langage HTML peut se prononcer « fin du bloc Directory ») concernent le répertoire `/usr/local/www_docs/personnel` et *tous ses sous répertoires* (sauf si un autre bloc modifie ces directives pour un de ces sous répertoires). La directive `<Location [...]>` joue un rôle similaire, mais elle fait référence à un chemin d'accès relatif à la racine du serveur et peut s'appliquer à un fichier au lieu d'un répertoire. `<Location /personnel/>` serait donc équivalent à la directive `<Directory>` mentionnée plus haut.

Pour définir les accès à l'ensemble du site, on indiquera le répertoire choisi pour recevoir les pages du site (`DocumentRoot`). Les directives proposées dans le fichier original sont :

`Options`

Parmi les options valides pour Apache citons :

- `Indexes`, qui précise que pour les requêtes dont l'URL concerne un répertoire au lieu d'un fichier, il faut envoyer la liste des fichiers du répertoire. En règle générale, on valide cette option pour les répertoires qui contiennent des documents techniques ou des programmes à télécharger afin de permettre aux utilisateurs d'utiliser le site comme s'il s'agissait d'un serveur FTP, mais on la retire dès que les pages du répertoire sont véritablement organisées et ordonnées.
- `FollowSymLinks`, selon laquelle il est permis de descendre dans une arborescence même s'il s'agit d'un lien symbolique (attention, un lien symbolique qui pointerait vers un répertoire système auquels les visiteurs du site ne doivent pas avoir accès constituerait une faille de sécurité flagrante).
- `ExecCGI`, qui indique que les programmes et scripts CGI contenus dans les mêmes répertoires que les pages peuvent être exécutés. On n'oubliera pas de valider cette option si on souhaite utiliser le handler `cgi-script`.
- `Includes`, qui permet que des pages fassent référence à des documents externes à inclure au moment du téléchargement. Tout comme les liens symboliques, cette possibilité peut présenter un certain danger si les documents insérés dans les pages contiennent des informations sur le système.

`AllowOverride`

Cette directive prend une des valeurs suivantes (on pourra se reporter à la documentation d'Apache pour plus de précisions).

- `None`
- `All`
- `AuthConfig`
- `FileInfo`
- `Indexes`
- `Limit`
- `Options`

Elle indique quelles options peuvent être substituées par de nouvelles valeurs contenues dans les fichiers `.htaccess`. Ces fichiers sont en quelque sorte des `access.conf` locaux qu'on place au sein même des répertoires afin d'éviter de reconfigurer le serveur lui-même à chaque ajout dans l'arborescence.

`order`, `allow`, `deny`

Ces directives pilotent le contrôle d'accès proprement dit au niveau Internet. Elles fonctionnent sur le même principe que les filtres disponibles dans les routeurs.

- `Order` prend la valeur `deny,allow` (attention, elle doit être écrite en un seul mot : sous la forme `deny, allow` elle serait refusée) ou `allow,deny`, respectivement pour indiquer que tout est interdit par défaut sauf ce qui est explicitement autorisé, ou l'inverse.
- `deny from` et `allow from` précisent quant à eux l'origine des connexions à refuser ou à autoriser. Il peut s'agir d'un ou plusieurs noms de machines ou de domaines, ou encore `none` ou `all`.

Premier démarrage

Nous allons avant tout libérer le port 80 pour le serveur, en retirant la référence au petit script décrit plus haut dans le fichier `inetd.conf`. Pour cela, il suffit simplement de mettre en commentaire ou d'effacer la ligne concernée et d'envoyer de nouveau le signal SIGHUP au démon `inetd`.

Si nous tentons de lancer le serveur depuis un compte utilisateur, nous obtenons ce type de message :

```
luc: /apache_1.1.1$ httpd -X -d .
bind: Permission denied
httpd: could not bind to port 80
luc: /apache_1.1.1$
```

Seul *root* a le droit de lancer des programmes qui « s'installent » sur un port dont le numéro est inférieur à 1024 (on parle de *port système*). Pour démarrer le serveur en tant qu'utilisateur autre que *root*, il faudra changer le numéro de port dans le fichier `httpd.conf` : par exemple, 8080 est souvent utilisé pour les serveurs qui tournent sous le nom d'un utilisateur quelconque.

On notera (figure 9.22) l'option `-X` qui permet de lancer Apache en mode débogage. Un seul processus est lancé, et il reste attaché au terminal depuis lequel il a été appelé : on peut donc l'interrompre très simplement. Les erreurs sont affichées à la console au lieu d'être écrites dans le fichier de *log*.

```
luc: /apache_1.1.1$ su
Password: ******
luc:/home/flg/apache_1.1.1$ httpd -X -d .
```

Figure 9.22 *Lancement du serveur Apache en tant que root*

En cas de problème de configuration

Si le maniement des fichiers de configuration s'avère problématique, on pourra essayer la configuration minimale proposée par défaut avant de l'affiner progressivement en y ajoutant d'autres options.

Mise en place permanente

La mise en place définitive impose de s'assurer que le serveur sera lancé automatiquement à chaque démarrage du système. Il est par ailleurs recommandé de mettre en place une *crontab* dont le rôle sera de surveiller le fonctionnement du serveur, en vérifiant que le processus est toujours actif, ou mieux, en demandant régulièrement une page HTML.

Bien entendu, le serveur doit être lancé sans l'option `-X`, puisqu'il ne doit être rattaché à aucun terminal. La surveillance des erreurs se fera par l'intermédiaire du fichier `error_log`.

Sous Linux, on pourra suivre la démarche suivante :

– Ajout de la commande démarrant le serveur dans les fichiers d'initialisation du système, par exemple dans le fichier `/etc/rc.d/rc.local`.

```
luc: $ su
Password: ********
luc: # vi /etc/rc.d/rc.local
[...]
/usr/local/httpd/httpd -d /usr/local/httpd/httpd
```

– Création du script de surveillance, ici `/etc/watch_httpd`. On utilise le contenu du fichier `httpd.pid` du répertoire `/usr/local/httpd/logs/` pour connaître le numéro de processus du serveur.

```
luc: # cat /etc/watch_httpd
#!/bin/sh

PIDFILE=/usr/local/httpd/logs/httpd.pid

if ! ps -p ‘cat $PIDFILE‘
then
  /usr/local/httpd/httpd -d /usr/local/httpd/httpd
fi
```

– Enregistrement du script dans la *crontab* de l'utilisateur *root* afin qu'il soit lancé à intervalles réguliers (ici toutes les 30 minutes, ce qui est sans doute insuffisant).

```
luc: # crontab -e
[...]
0,30 * * * * /etc/watch_httpd >/dev/null 2>&1
[...]
```

9.3.1 Utilisation des modules

Certaines fonctions disponibles avec Apache ne sont pas directement intégrées au serveur, mais proposées sous forme de modules à rattacher lors de la compilation (sur certains systèmes, les dernières versions d'Apache permettent également d'utiliser des modules *dynamiques*, c'est-à-dire compilés indépendamment et chargés au besoin lors de l'exécution).

Nous verrons section 10.3.3 page 366 comment accéder aux fonctions complémentaires du serveur Apache, en l'occurence à travers l'utilisation du module `imap`.

L'ajout d'un module se fait au moment de la compilation du serveur. Le module doit tout d'abord être indiqué dans le fichier de configuration, sous une forme analogue à :

```
Module imap_module mod_imap.o
```

Cette ligne précise le nom du module, suivi du nom du fichier objet à inclure lors de l'édition des liens. Ici, il s'agit du module `imap`, qui gère les cartes cliquables au niveau du serveur (voir la section 10.3.3 page 360).

Pour ajouter le module `imap`, il suffit donc d'insérer cette ligne dans le fichier de paramètres `Configuration` (elle y figure d'ailleurs par défaut), puis de relancer la commande `Configure` et de recompiler le serveur avec `make`.

La plupart des modules recherchent des paramètres de fonctionnement spécifiques dans les fichiers de configuration du serveur (`httpd.conf`...) ; par exemple, pour utiliser le module `imap`, il faudra associer le module aux fichiers portant l'extension `.map` en insérant dans le fichier `srm.conf` la ligne :

```
AddHandler imap-file map
```

9.3.2 Notion d'adresse virtuelle (virtual host)

Il est parfois nécessaire de mettre en place plusieurs sites Web sur la même machine. Si cette dernière n'est pas particulièrement 'gonflée', il peut être préférable de n'utiliser qu'un seul et même serveur pour tous ces sites. Il faut alors disposer de plusieurs adresses IP, donc plusieurs interfaces, et indiquer au serveur qu'il doit 'surveiller' le port 80 (ou un autre) sur chacune de ces interfaces.

La création d'une interface virtuelle, qui consiste à affecter une adresse IP supplémentaire à une interface physique existante (par exemple une carte ethernet), n'est pas encore possible sur tous les systèmes. Sous Solaris ou Linux, elle s'effectue grâce à la commande `ifconfig`, en utilisant successivement l'adresse de l'interface suivie des labels `:0`, `:1`, etc. (figure 9.23 page 306). Bien sûr, l'adresse doit être routée.

Une fois l'interface créée, il faut préciser au serveur sur quelles adresses les requêtes doivent parvenir.

L'option `BindAdress` du fichier de configuration `httpd.conf` permet de préciser à quelle adresse le serveur se trouve, lorsqu'il n'y en a qu'un. Un astérisque demande au serveur d'écouter sur *toutes* les interfaces. Ici, ce n'est évidemment pas souhaitable. Il faudra donc préciser l'adresse de l'interface par défaut ; c'est à elle que s'appliqueront les options courantes.

Pour paramétrer les interfaces supplémentaires, on utilisera la directive `<VirtualHost>`, avec la syntaxe suivante.

Ici l'adresse IP correspondant au nom `www2.fenetre.fr` est `192.168.22.35`, et l'entrée correspondante a été ajoutée au DNS.

```
<VirtualHost home.fenetre.fr>
  ServerAdmin webmaster@home.fenetre.fr
  DocumentRoot /www/docs/home
  ServerName home.fenetre.fr
  ErrorLog logs/home.error_log
  TransferLog logs/home.access_log
</VirtualHost>
```

La plupart des options de configuration peuvent être dupliquées au sein d'un bloc de ce type, à l'exception bien entendu des options relatives au fonctionnement général du serveur (`ServerRoot`, `MaxSpareServers`, etc...).

Ici nous avons configuré un second site. Nous disposons donc maintenant de deux serveurs Web, dont les URL sont respectivement :

```
http://www.fenetre.fr
http://home.fenetr.fr
```

Note : le serveur Apache permet également de configurer plusieurs serveurs sans faire appel à des interfaces virtuelles, en utilisant un champ particulier des en-têtes HTTP : le champ `Host`. On peut ainsi créer plusieurs sites à partir du même serveur et sur la même adresse IP, après avoir ajouté les entrées CNAME correspondantes dans le DNS.

Cette technique, dite des Non-IP Virtual Hosts, n'est pas recommandée, car seuls les navigateurs très récents renseignent le champ `Host`. Tous les autres navigateurs renverront automatiquement vers le site par défaut.

9.4 Installation de Netscape Commerce Server

L'installation du serveur de Netscape est nettement plus simple que celle d'Apache. Toutes les opérations sont pilotées par l'intermédiaire d'un navigateur, qui fournit une interface somme toute comparable aux fenêtres et boîtes de dialogue d'un logiciel classique sous Windows ou MacOS. Tout comme pour le serveur Apache, nous ne verrons que la version Unix du logiciel, mais Netscape fonctionne également sous Windows NT.

Les fonctionnalités offertes par le serveur commercial de Netscape sont comparables à celles d'Apache. Tout comme ce dernier, il peut en outre être enrichi par des modules (ou *plug-ins*) externes, en particulier grâce à une interface standardisée (NS-API). Netscape propose un kit de développement pour exploiter son API.

```
luc:~# ifconfig eth0:0 192.168.22.35
luc:~# ifconfig -a

eth0      Link encap:10Mbps Ethernet  HWaddr 00:A0:24:B3:07:02
          inet addr:192.168.22.34  Bcast:192.168.22.31  Mask:255.255.255.224
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:596219 errors:0 dropped:0 overruns:0
          TX packets:246764 errors:0 dropped:0 overruns:0
          Interrupt:10 Base address:0x300

eth0:0    Link encap:10Mbps Ethernet  HWaddr 00:A0:24:B3:07:02
          inet addr:192.168.22.35  Bcast:192.168.22.31  Mask:255.255.255.224
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0

luc:~# route add -host 192.168.22.35 eth0:1
```

Figure 9.23 *Configuration d'une interface virtuelle*

9.4.1 Lancement de l'installation

Avant tout, l'archive doit être récupérée depuis le site Web de Netscape (version de démonstration) ou sur le CD-Rom du produit. La version de démonstration se présente sous la forme d'un fichier archive compressé que nous éclaterons dans un répertoire temporaire, ici `/home/ns-install`.

L'installation proprement dite se lance à l'aide de la commande `ns-setup`, qui se trouve dans le sous-répertoire de l'archive.

```
/home/ns-install/export/https/install/ns-setup
```

La figure 9.24 montre le démarrage de l'installation.

```
# ./ns-setup

Netscape Communications Corporation
Netscape Commerce Server QuickStart installation.

[…]
Full name [gide.fenetre.fr]: gide.fenetre.fr

Using hostname gide.fenetre.fr, port 14087.

[…]

If you want or need to use a PC, Macintosh, or other remote system, enter
NONE here, and open the URL http://gide.fenetre.fr:14087/
with your forms-capable PC or Macintosh network navigator.
Network navigator [netscape]:
Attempting to run: netscape http://gide.fenetre.fr:14087/ \&
```

Figure 9.24 *Lancement de l'installation de Netscape Commerce Server*

Le nom de machine demandé lors de l'installation est celui du serveur. Ainsi que nous l'avons vu dans les paragraphes précédents, il ne s'agit pas nécessairement de celui de la machine, mais souvent d'un équivalent au sens DNS CNAME. L'installation présentée ici concerne le serveur interne de la société FeNETre, qui se trouve sur `gide`. Le nom demandé par Netscape représente le `ServerName` évoqué pour Apache : il serait donc important, dans le cas d'un serveur public, d'indiquer son nom officiel (par exemple `www.fenetre.fr`) au lieu de son nom au sein du réseau local. Dans tous les cas, le premier écran de configuration permet de modifier le nom indiqué, on peut donc se contenter de valider le nom proposé par défaut s'il correspond à un des noms de la machine.

Le programme d'installation démarre un serveur HTTP sur un port libre choisi au hasard et dont le numéro exact est précisé par le programme d'installation. Toute la suite de la configuration se déroule donc par l'intermédiaire d'une interface HTML. Si on dispose d'un client HTML sur la machine elle-même (ainsi que d'un environnement graphique tel X Window ou OpenWin), on peut continuer l'installation depuis cette dernière ; dans le cas contraire il faut se connecter sur le port indiqué à l'aide d'un navigateur depuis une autre machine, par exemple un poste sous Windows ou MacOS. Le programme d'installation attend un nom de commande à exécuter pour lancer le navigateur sur le serveur. Par défaut il s'agit du client de *Netscape*, mais on peut bien sûr utiliser `Mosaic` ou tout autre navigateur capable d'afficher des formulaires (ce qui est le cas de tous les navigateurs récents). En mode texte, un client spécial tel que `lynx` peut faire l'affaire, mais la saisie des informations risque d'être laborieuse.

Les différents écrans de la phase de configuration sont largement commentés, nous allons passer les principaux en revue en insistant sur les paramètres fondamentaux. Une fois le navigateur lancé, il suffit de cliquer sur le bouton `"Start the installation"` pour passer au menu général du programme de configuration. Il est également possible de mettre à jour une version déjà installée : dans ce cas, on pourra conserver les paramètres de l'ancienne configuration.

La configuration se fait en trois écrans, qu'on pourrait comparer aux trois fichiers de configuration du serveur Apache.

L'écran de configuration du serveur lui-même (figure 9.28 page 311) permet de définir :

- le nom du serveur (au sens DNS : c'est le `ServerName` d'Apache) ;
- son adresse IP (`BindAddress`) ;
- le port sur lequel il doit s'installer ;
- le répertoire dans lequel il doit placer ses fichiers (le répertoire `ServerRoot`), par défaut `/usr/ns-home` ;
- l'identificateur sous lequel il doit fonctionner : la plupart du temps, il s'agira de l'utilisateur *nobody*, mais on peut préférer créer un utilisateur factice particulier (voir la section 9.2.3 page 292) ;
- le nombre de processus lancés au démarrage.

Figure 9.25 *Écran d'accueil de l'installation du serveur Netscape*

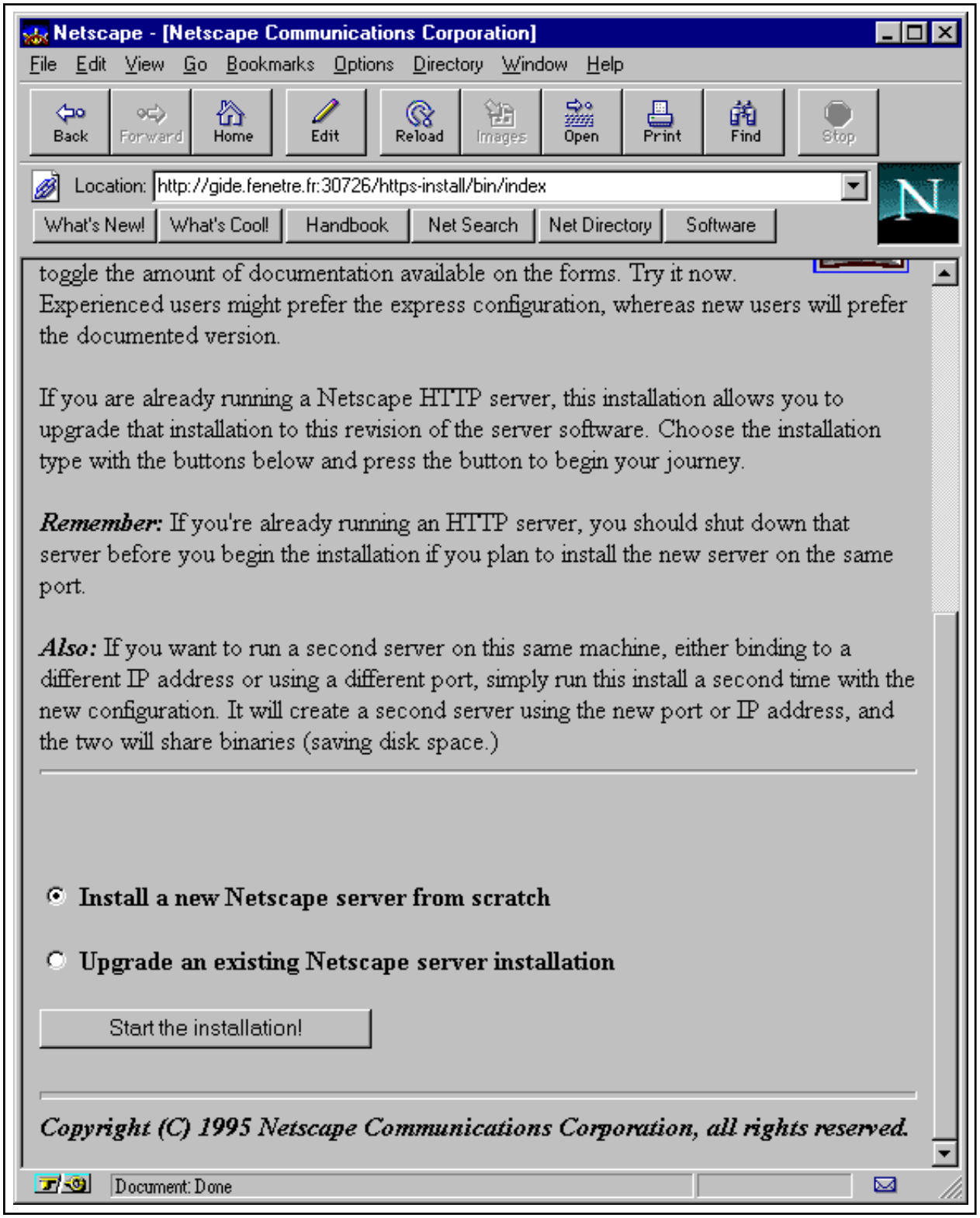

Figure 9.26 *Démarrage de l'installation ou de la mise à jour*

Figure 9.27 *Menu principal de l'installation*

Figure 9.28 *Configuration des options propres au serveur*

L'écran de configuration du site (figure 9.29 page suivante) se rapporte aux options de mise en place des documents :

- le répertoire du site, analogue au `DocumentRoot` d'Apache ;
- le nom des fichiers d'index (voir `DirectoryIndex`), c'est-à-dire les pages HTML qui doivent être envoyées en réponse aux requêtes portant sur un répertoire et non sur un fichier : la plupart du temps, il s'agit bien sûr de `index.html`, mais on peut en indiquer plusieurs en les séparant par des virgules ;
- le type de page fabriquée par le navigateur lorsqu'il n'y a pas de fichier d'index (voir `FancyIndexing`) : il s'agit dans tous les cas de la liste des fichiers du répertoire, elle peut être 'décorée' par des icones ou se présenter comme une liste simple ;
- la référence de la page d'accueil, qui peut se trouver ailleurs qu'à la racine du répertoire du site.

Une particularité du serveur de Netscape est que son administration s'effectue par l'intermédiaire d'un navigateur Web, tout comme son installation. Le troisième écran de configuration

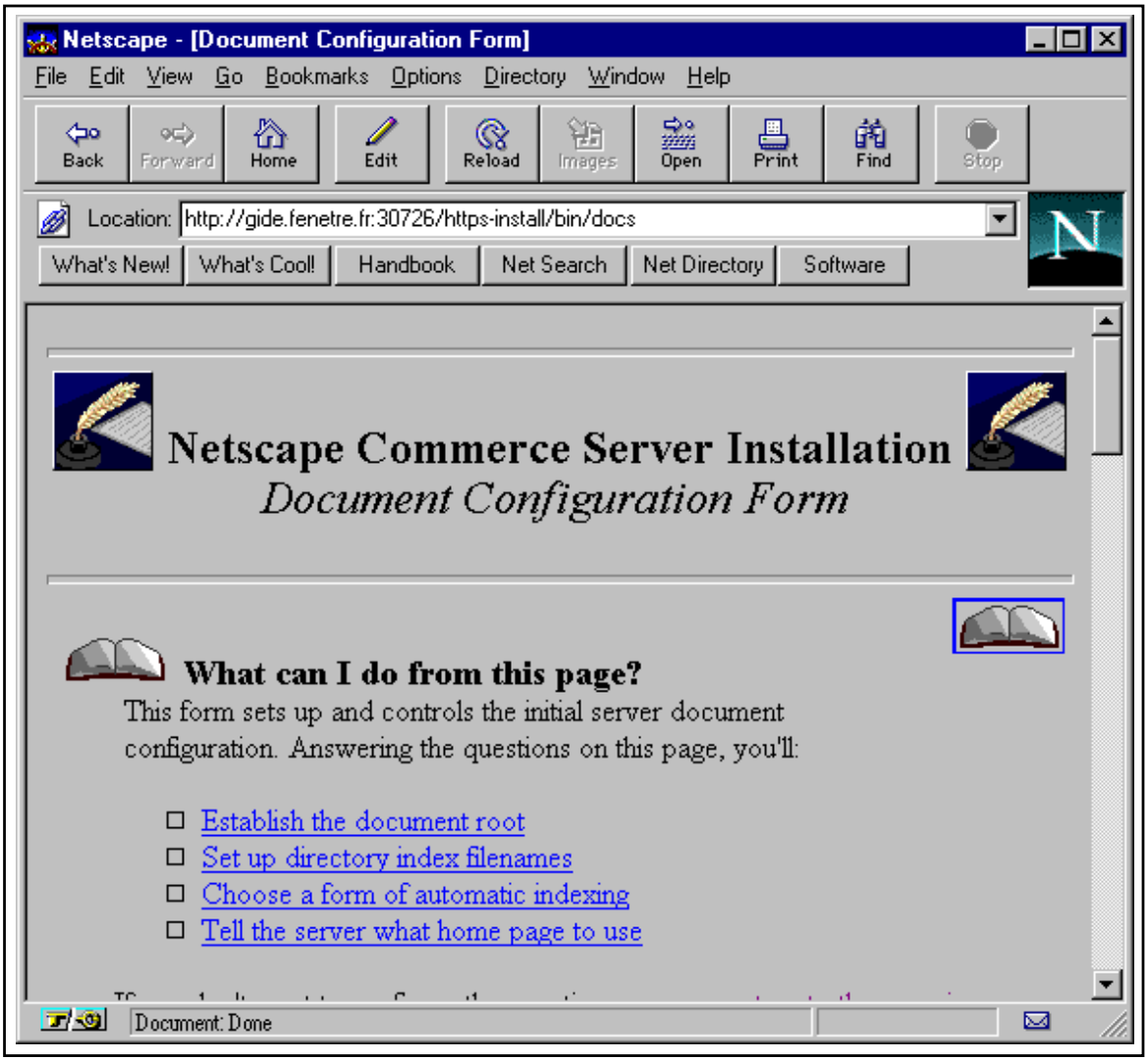

Figure 9.29 *Configuration des options du site Web*

(figure 9.30 page ci-contre) concerne précisément le serveur d'administration. On y précisera :

- le nom et le mot de passe de l'administrateur ;
- la liste des machines qui peuvent accéder au serveur d'administration ;
- le port sur lequel il doit se mettre en veille (on pourra par exemple choisir le port système 81) ;
- l'identificateur sous lequel il doit fonctionner – en ce qui concerne le serveur d'administration, il n'est généralement pas dangereux de le laisser fonctionner sous l'identificateur *root*, puisque son accès est restreint.

Une fois la configuration terminée, un écran intermédiaire résume les options choisies (figure 9.31 page 314). Si elles sont convenables, l'installation peut continuer.

Le serveur d'administration (figure 9.32 page 315) comporte une liste très complète de para-

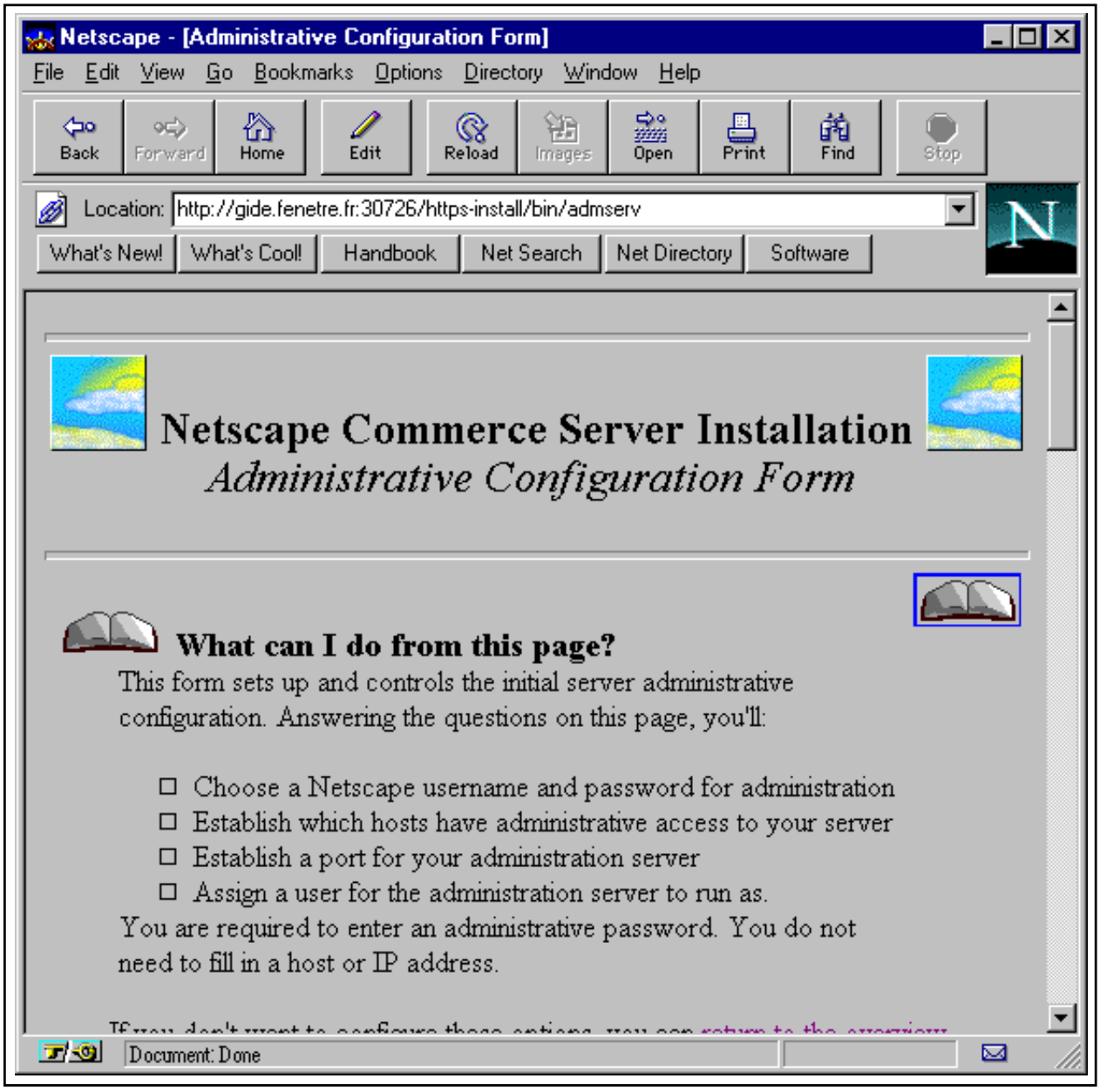

Figure 9.30 *Configuration du serveur d'administration*

mètres, qui définissent le comportement du serveur au niveau des documents, mais aussi de la sécurité, de l'authentification, voire du cryptage pour les transactions commerciales. On s'y connecte par l'intermédiaire du port défini plus haut, puis en entrant le nom et le mot de passe de l'administrateur. Le détail des options d'administration du serveur Netscape dépasse le cadre de cet ouvrage ; on se rapportera à la documentation papier.

Note : tout comme Apache, Netscape sait gérer plusieurs serveurs sur la même machine (*virtual hosts*). Il est donc possible de configurer le serveur pour écouter sur plusieurs adresses différentes, si le système le permet.

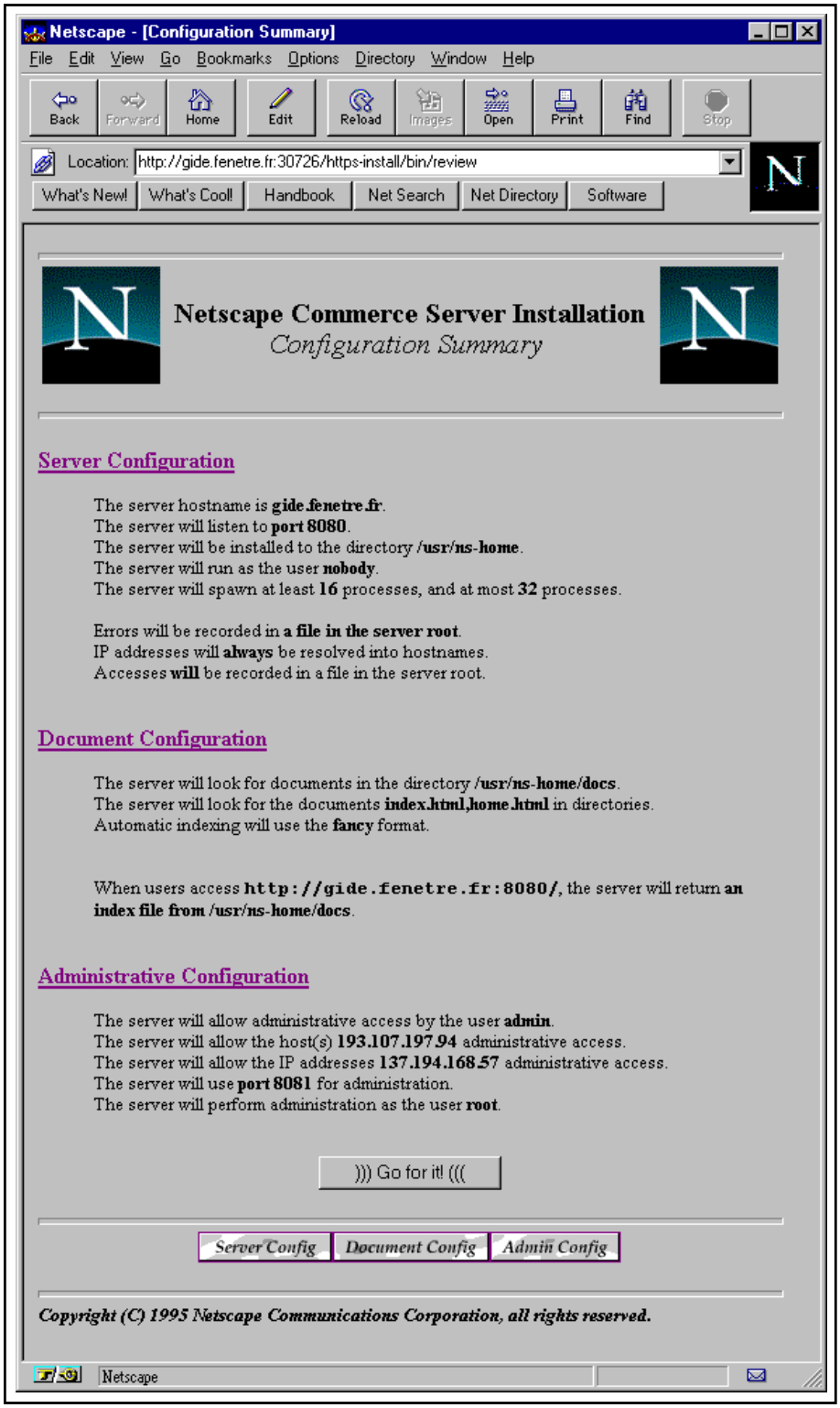

Figure 9.31 *Résumé des options de configuration*

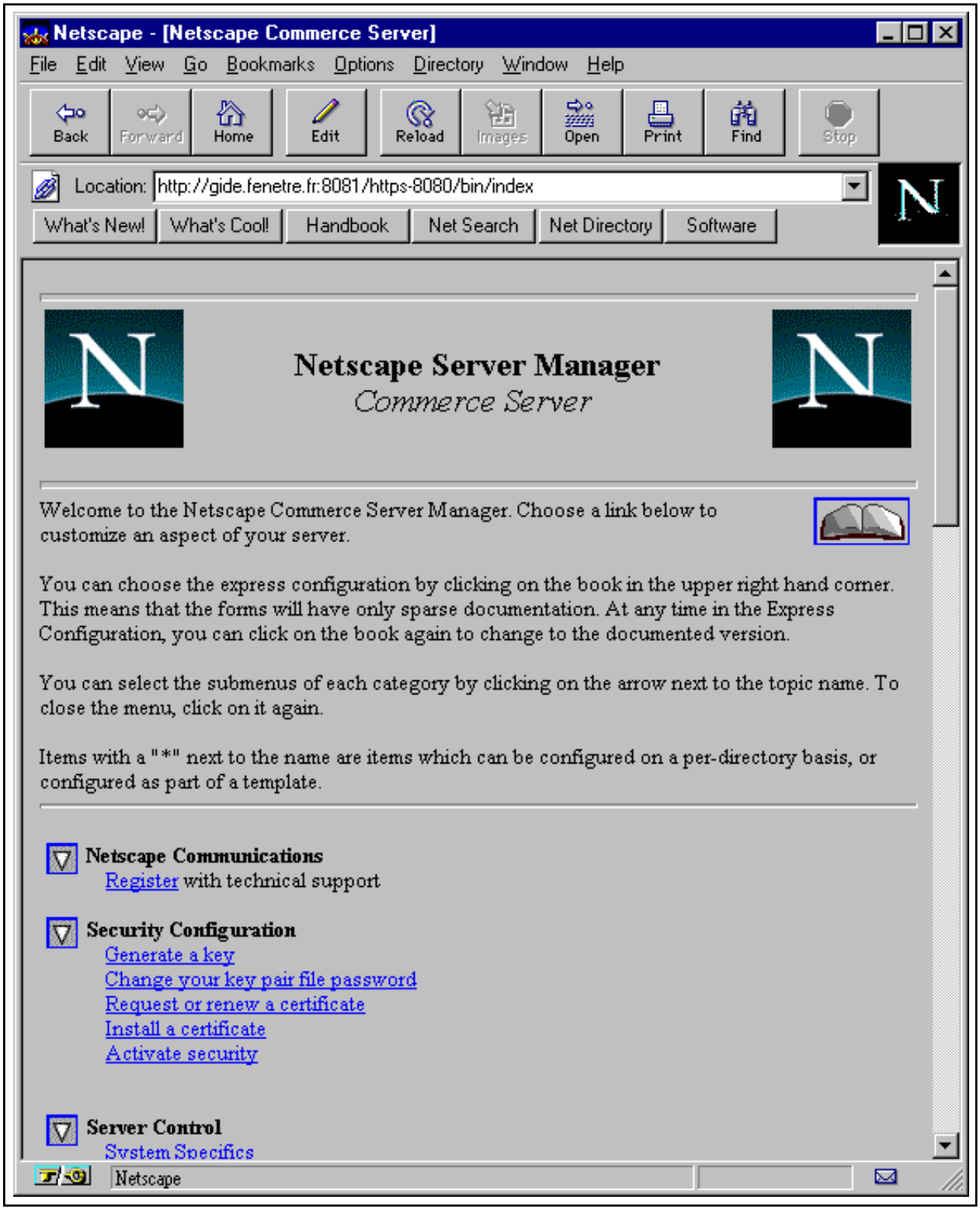

Figure 9.32 *Menu général du serveur d'administration*

9.5 Mise en place du contenu

Une fois le serveur correctement installé, il s'agit de mettre en place les documents. Le chapitre suivant présente les divers éléments utilisés lors de la fabrication des pages : le langage HTML d'une part, mais aussi les différents programmes et scripts permettant de traiter des formulaires, de personnaliser les pages, d'automatiser certaines fonctions, ou même d'accéder à des bases de données.

D'une manière formelle, la mise en place du contenu consiste essentiellement à copier les différents fichiers (pages, images, scripts, voire sons ou vidéos) à leurs places respectives : le répertoire des documents (`DocumentRoot`) pour les pages et fichiers de données, le répertoire des scripts (`ScriptAlias`) pour les programmes, à moins qu'ils ne soient placés aux mêmes endroits que les documents eux-mêmes.

Il est cependant indispensable de penser à effectuer certains contrôles visant à renforcer la stabilité et la sécurité du serveur : droits d'accès, liens symboliques, droits en exécution et propriétaires des programmes... Nous allons établir une rapide liste des étapes techniques à suivre et des points à vérifier avant d'ouvrir le serveur au public.

9.5.1 Check-list de l'installation

– Vérifier que le nom du serveur correspond bien à une entrée valable dans le DNS, qu'il s'agisse d'un champ d'adresse (A) ou d'un alias (CNAME).

Direct (`named.hosts`)
```
gide    IN      A       192.168.22.34
www     IN      CNAME   gide
```
Reverse (`named.rev`)
```
34      IN      PTR     gide.fenetre.fr
```

– Vérifier que le nom indiqué dans la configuration du serveur (`ServerName`) est bien le nom officiel s'il est différent du nom de la machine.

Configuration Apache (`httpd.conf`)
```
ServerName      www.fenetre.fr
```

– Vérifier que les chemins d'accès sont corrects et que le répertoire des scripts a été défini le cas échéant.

Configuration Apache (`httpd.conf`)
```
ServerRoot      /usr/local/httpd
DocumentRoot    /usr/local/www_docs
```
Configuration Apache (`httpd.conf`)
```
ScriptAlias     /cgi-bin/ /usr/local/httpd/cgi-bin/
```

– Ajouter les commandes nécessaires au lancement automatique du serveur au démarrage de la machine.

  En mode `inetd`, insérer une entrée dans le fichier `/etc/inetd.conf`. Par exemple :

```
www  stream  tcp  nowait  root  /usr/local/httpd/httpd  httpd
```

  En mode `standalone`, ajouter la commande de démarrage du serveur aux fichiers d'initialisation (`rc.*`).

  Sous Linux (`/etc/rc.d/rc.inet2` ou `rc.local`) :

```
/usr/local/httpd/httpd -d /usr/local/httpd/httpd
```

– Mettre en place les documents, en veillant à respecter les noms de fichiers. Attention aux minuscules/majuscules !
– Vérifier le fonctionnement du serveur et des scripts.
– Vérifier que les scripts ne font pas appel à des commandes du système dont les paramètres seraient passés par formulaire : il s'agirait d'une faille flagrante dans la sécurité du système.
– Repérer les liens symboliques et s'assurer qu'ils ne permettent pas de *remonter* indûment dans l'arborescence du disque.
– Ne laisser aucun fichier appartenant à des utilisateurs ou groupes ayant des droits particuliers sur la machine (*root*, *bin*...).

  Si possible, créer par exemple un utilisateur *www* et un groupe *wwwadm* pour tous les documents du site. Plusieurs personnes différentes pourront ainsi accéder aux documents, éventuellement avec des droits différents pour le groupe.

– Vérifier que le serveur fonctionne bien avec le minimum de droits.

  Sauf si les scripts et programmes doivent accéder à des informations ou des ressources en écriture, le serveur devrait fonctionner sous *nobody/nogroup* (généralement *-1/-1*). Sinon, utiliser le compte *www* (voir plus haut) ou un autre compte *ad hoc*.

10

Le langage HTML et son utilisation

Les pages consultées sur le Web sont rédigées dans un langage de description mis au point au CERN et dont les objectifs originaux étaient d'une part d'homogénéiser la présentation des documents mis en ligne, d'autre part de permettre la création de documents distribués sur plusieurs serveurs distincts, grâce à la notion de lien hypertexte. Les niveaux 2 et 3 du standard HTML (Hypertext Markup Language) y ajoutent des notions de formatage évolué dont l'utilisation s'éloigne parfois sensiblement de la vocation première du langage lui-même, mais qui permettent d'obtenir des effets de présentation de plus en plus comparables à ceux permis par les traitements de texte évolués.

De nombreux ajouts sont régulièrement proposés par les sociétés les plus actives dans la mise au point des navigateurs, en l'occurrence Netscape et Microsoft. Certains correspondent à de véritables besoins, d'autres ne sont que des gadgets marketing ; dans tous les cas, c'est le **Web Consortium**, un organisme paritaire qui se réunit régulièrement, qui statue sur les éléments qui seront intégrés au standard. Après une période assez confuse durant laquelle les différentes versions des navigateurs du marché ont proposé des fonctions de plus en plus hétéroclites, la tendance actuelle est à la simplification et à l'uniformisation, afin de redonner au HTML son véritable de rôle de standard de diffusion documentaire.

Le but de ce chapitre n'est pas de faire une présentation détaillée du langage et de son utilisation, mais uniquement de présenter les éléments indispensables pour commencer à rédiger quelques documents, et de décrire les différents procédés utilisés dans le traitement des formulaires et la mise en place de programmes destinés à générer des pages à la volée. Les paragraphes sont illustrés par une série d'exemples qui correspondent aux différentes étapes de la conception d'un premier site.

10.1 Généralités

Le langage HTML s'appuie sur des instructions de formatage placées au sein du texte pour afficher un document en interprétant ces instructions (ou *tags*) selon, d'une part, ce que le navigateur est capable de faire, d'autre part la manière dont il est configuré. Il est intéressant de constater qu'on ne peut jamais présumer de l'aspect exact qu'un navigateur Web donnera à un document. En effet, il est généralement permis à l'utilisateur de choisir la forme et la taille de la police par défaut, de supprimer les images, d'imposer des couleurs pour le fond et le texte, etc.

Parmi les autres facteurs qui influencent l'aspect du document à l'affichage, citons la résolution de l'écran, la taille de la fenêtre du navigateur (le texte s'adapte généralement à la largeur de la fenêtre, sauf s'il a été volontairement délimité par un tableau), le nombre et la palette des couleurs affichables.

Un des principes fondamentaux du langage est que tout document doit pouvoir transiter sur n'importe quel réseau même si les caractères sont encodés sur 7 bits. Cela signifie que les caractères de contrôle ainsi que tous les caractères accentués et spéciaux doivent être proscrits. Dans la pratique, le jeu de caractères `iso-latin1` est désormais reconnu, ce qui permet d'éviter l'encodage des caractères accentués. En revanche, une autre conséquence majeure de cette restriction se rapporte à la mise en page du code HTML lui-même. Ainsi que nous l'avons vu lorsque nous évoquions les transferts de fichiers par *email*, les sauts de lignes doivent être encodés au même titre que les autres caractères de contrôle. De fait, en HTML, les sauts de ligne représentés dans le code ne correspondent pas à des sauts de ligne dans la mise en page : ils sont tout simplement ignorés et remplacés par un espace, tout comme les espaces multiples.

Ainsi le texte :

```
Plus loin dans la vallée se
trouve une          cahute. Pas de
vitres aux fenêtres, pas
de serrure aux portes, chacun peut y entrer, s'y

plaire,
et                y dormir.
```

ou encore :

```
Plus loin dans la vallée se trouve une cahute.

Pas de vitres aux fenêtres, pas de serrure aux portes,
chacun peut y entrer, s'y plaire, et y dormir.
```

donnera de toute façon à l'affichage :

On peut en particulier utiliser cette particularité du standard pour formater les sources des documents HTML d'une manière plus lisible, en ajoutant des retraits marquant la hiérarchie des

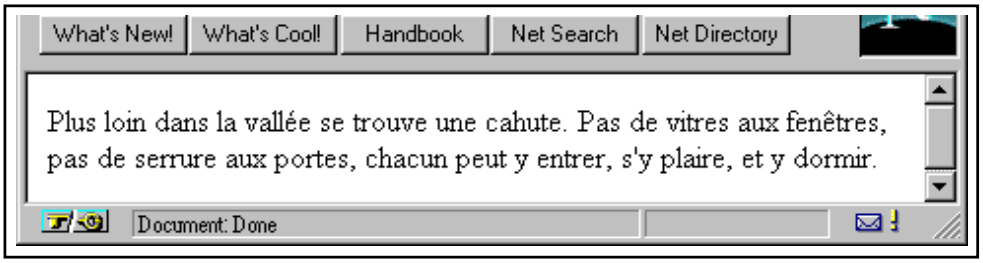

Figure 10.1 *Les navigateurs ignorent les espaces multiples et les sauts de lignes*

instructions imbriquées, tout comme dans un programme informatique classique. On notera d'ailleurs qu'il n'est pas indispensable de faire tenir les instructions HTML sur une ligne. Il est permis de couper les instructions n'importe où entre les paramètres :

```
<BODY
BGCOLOR=red
>
```

Les instructions HTML (*tags* HTML) se notent sous la forme <INSTRUCTION> et se *ferment* le cas échéant par la notation négative </INSTRUCTION>. Les instructions de mise en forme du texte qui encadrent généralement un groupe de mots ou un paragraphe doivent être fermées, tandis que certaines autres destinées à insérer un élément (une image, par exemple) comportent une seule instruction.

Certaines fonctions acceptent un ou plusieurs paramètres, qui se placent au sein même de l'instruction. Ainsi <IMG SRC="nouveau.gif" WIDTH=100 HEIGHT=200> signifie « insérer l'image dont l'URL relatif est `nouveau.gif` en lui donnant les dimensions 100x200 ».

Les instructions peuvent être rédigées indifféremment en minuscules ou en majuscules. En revanche, les noms de fichiers ou les URL auquels elles font parfois référence doivent être traités comme des noms de fichiers ou URL normaux, c'est-à-dire que si le serveur fonctionne sous un système de type Unix, la forme exacte du chemin d'accès et du nom de fichier doit être respectée.

Les navigateurs récents gèrent assez correctement les instructions de formatage imbriquées. Cependant, une bonne partie de ceux encore en fonction ne le font pas, par conséquent il vaut mieux s'assurer que dans les deux cas, le résultat sera lisible.

10.1.1 Les niveaux du standard

On distingue pour l'instant trois versions majeures du langage. La première permet de présenter un document de manière structurée grâce à six niveaux de titres, des instructions définissant le style des caractères, la possibilité d'insérer des listes et celle de mettre en place des champs de saisie dans des formulaires. La seconde y ajoute principalement la capacité

de formater des tableaux, et la troisième celle de disposer le texte autour des images. Ces versions sont totalement compatibles, puisque les navigateurs se contentent d'ignorer purement et simplement les instructions qu'ils ne connaissent pas. En revanche, des différences sensibles au niveau de la mise en page peuvent apparaître. C'est pour cette raison qu'il est parfois préférable de mettre au point deux jeux de pages utilisant respectivement les instructions *standards* et des instructions *avancées*.

Les versions des navigateurs utilisés en France sont souvent disparates, et il est parfois indispensable de simplifier un site si on ne souhaite pas en gérer deux versions différentes.

10.2 Syntaxe

10.2.1 Encodage HTML

Le premier niveau du standard HTML définit une série de codes spéciaux ou entités (*entities*) destinés à représenter par une série de caractères du jeu ASCII standard les principaux caractères accentués et spéciaux des langues occidentales (il s'agit en fait de la partie supérieure du jeu `iso-latin1`). Chaque code est précédé du signe `&` et suivi du signe `;` ce qui impose bien entendu de représenter le signe `&` lui-même par un code particulier. Les signes inférieur `<` et supérieur `>` et les guillemets `"` sont également encodés, car ils servent au sein des instructions.

On remarque que chaque entité (tableau 10.1 page suivante) peut être représentée indifféremment par son nom ou par son code numérique précédé du signe #. Ainsi `&eacute;` et `&#233;` représentent le caractère é.

Cependant cette possibilité ne devrait pas être utilisée, car elle ôte toute généricité au système d'encodage. Il est possible de représenter l'entité é à partir de n'importe quel jeu de caractères occidental, quitte à afficher un simple e à la place, et ce sans nécessairement connaître la structure du jeu `iso-latin1`; en revanche, afficher la même entité à partir de son code numérique nécessite de disposer d'une table de conversion du jeu `iso-latin1` vers le jeu de caractères du système.

On notera l'importance de la distinction majuscules/minuscules dans les entités, tout comme dans les noms de fichiers, puisqu'il est possible d'accentuer des voyelles majuscules.

10.2.2 Les différentes parties d'un document

En règle générale, un document HTML se compose d'un en-tête (HEAD), qui n'est pas affiché mais sert à fournir des informations complémentaires au navigateur ou à d'éventuels moteurs de recherche, et d'un corps (BODY), qui contient le texte à mettre en page.

Les informations qu'on retrouve le plus souvent dans l'en-tête sont le titre du document,

Entité		Code	Signification						
&amp;	&								
&quot;	"								
&lt;	<								
&gt;	>								
		160	Espace insécable	&Agrave;	À	192	&agrave;	à	224
&iexcl;	¡	161	Point d'exclamation inversé	&Aacute;	Á	193	&aacute;	á	225
&cent;	¢	162		&Acirc;	Â	194	&acirc;	â	226
&pound;	£	163		&Atilde;	Ã	195	&atilde;	ã	227
&curren;	¤	164		&Auml;	Ä	196	&auml;	ä	228
&yen;	¥	165		&Aring;	Å	197	&aring;	å	229
&brvbar;	¦	166		&AElig;	Æ	198	&aelig;	æ	230
&sect;	§	167	Saut de section	&Ccedil;	Ç	199	&ccedil;	ç	231
&uml;	¨	168	*Umlaut*	&Egrave;	È	200	&egrave;	è	232
&copy;	©	169	Copyright	&Eacute;	É	201	&eacute;	é	233
&ordf;	ª	170		&Ecirc;	Ê	202	&ecirc;	ê	234
&laquo;	«	171		&Euml;	Ë	203	&euml;	ë	235
&not;	¬	172		&Igrave;	Ì	204	&igrave;	ì	236
&shy;	-	173	Tiret demi-cadratin	&Iacute;	Í	205	&iacute;	í	237
&reg;	®	174	*Registered*	&Icirc;	Î	206	&icirc;	î	238
&macr;	¯	175	Surlignage	&Iuml;	Ï	207	&iuml;	ï	239
&deg;	°	176	Degré	&ETH;	Ð	208	&eth;	ð	240
&plusmn;	±	177	Plus ou moins	&Ntilde;	Ñ	209	&ntilde;	ñ	241
&sup2;	²	178	Carré	&Ograve;	Ò	210	&ograve;	ò	242
&sup3;	³	179	Cube	&Oacute;	Ó	211	&oacute;	ó	243
&acute;	´	180	Accent aigu	&Ocirc;	Ô	212	&ocirc;	ô	244
&micro;	µ	181		&Otilde;	Õ	213	&otilde;	õ	245
&para;	¶	182	Saut de paragraphe	&Ouml;	Ö	214	&ouml;	ö	246
&middot;	·	183	Point mi-hauteur	&times;	×	215	&divide;	÷	247
&cedil;	¸	184	Cédille	&Oslash;	Ø	216	&oslash;	ø	248
&sup1;	¹	185	Puissance 1	&Ugrave;	Ù	217	&ugrave;	ù	249
&ordm;	º	186		&Uacute;	Ú	218	&uacute;	ú	250
&raquo;	»	187		&Ucirc;	Û	219	&ucirc;	û	251
&frac14;	¼	188	Quart	&Uuml;	Ü	220	&uuml;	ü	252
&frac12;	½	189	Demi	&Yacute;	Ý	221	&yacute;	ý	253
&frac34;	¾	190	Trois quarts	&THORN;	Þ	222	&thorn;	þ	254
&iquest;	¿	191	Point d'interrogation inversé	&szlig;	ß	223	&yuml;	ÿ	255

Tableau 10.1 *Entités HTML*

délimité par l'instruction <TITLE> et son inverse </TITLE>, l'URL original du document, et des renseignements divers à l'usage des moteurs de recherche ou du navigateur délimités par l'instruction <META>.

L'instruction BASE permet de préciser l'URL où l'on peut trouver le document. Elle est utile lorsque les instructions utilisées par la suite dans la page font référence à des URL relatifs. Lorsqu'on enregistre le code source d'une page récupérée sur le Net, on ne peut généralement pas retrouver les images par la suite, sauf en les récupérant une par une. Si l'URL de base du document a été précisé dans la page, le navigateur pourra télécharger les images depuis le

site d'origine. L'inconvénient de cette pratique, si elle n'est pas automatisée par programme, est qu'elle complique les déplacements de fichiers au sein de l'arborescence du serveur.

En dépit de son nom, le *titre* du document n'est pas affiché avec le document lui-même. Il est en général précisé dans la barre supérieure de la fenêtre du navigateur.

```
<HTML>
<HEAD>
<TITLE>
TITRE DU DOCUMENT
</TITLE>
<BASE HREF="http://www.fenetre.fr/html/tutorial/head.html">
</HEAD>
<BODY>
Le titre apparaît en haut de la fenêtre
</BODY>
</HTML>
```

Figure 10.2 *Utilisation de l'instruction <TITLE>*

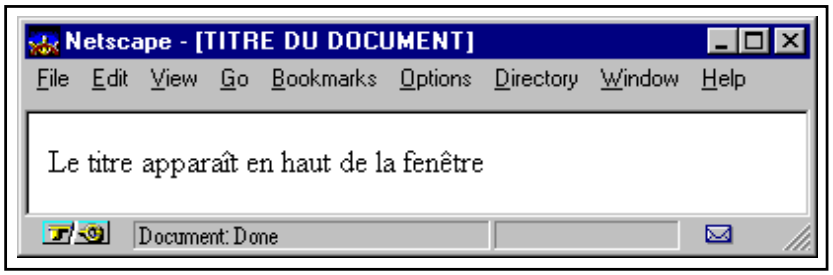

Figure 10.3 *Utilisation de l'instruction <TITLE>*

Les champs `META` sont principalement utilisés pour indiquer une liste de mots clés auxquels la page fait référence, ou encore un résumé du contenu de cette dernière. Cette information est récupérée par certains moteurs de recherche qui s'en servent pour indexer les documents dans leurs listes thématiques.

L'instruction <BODY> indique le début du corps du document. Aucune des instructions <HTML>, <HEAD> ou <BODY> n'est réellement obligatoire pour les navigateurs. Cependant, leur présence systématique permet au besoin d'automatiser de nombreuses manipulations à l'aide de programmes très simples : elle est donc souhaitable. La plupart des éditeurs HTML disponibles sur le marché veillent d'eux-mêmes à ce que ces instructions soient présentes (ils ajoutent d'ailleurs très souvent un champ META nommé `GENERATOR` dans l'entête du document, afin de préciser le nom et la version du produit ayant servi à composer les pages ; par exemple <META NAME="GENERATOR" CONTENT="Internet Assistant for Microsoft Word 2.0z">)

10.2.3 Affichage ou enregistrement des documents source

La méthode la plus efficace pour apprendre à formater des pages HTML consiste sans aucun doute à explorer les documents existants et à consulter leur code source. Ce dernier est nécessairement disponible, puisque le navigateur s'en sert pour afficher la page. D'une certaine manière, on pourrait dire que sur le Web, le travail de chacun est accessible à tous, d'autant que les navigateurs offrent maintenant la possibilité d'enregistrer également les images.

Bien entendu, tout ce qui circule sur le Net est soumis aux mêmes lois sur les droits de diffusion et de copie que n'importe quel autre document ou création. Cependant, si le plagiat pur et simple est légalement discutable, moralement douteux et pratiquement mal accepté par la communauté des utilisateurs de l'Internet, l'usage veut qu'on puisse librement *s'inspirer* de ce qui existe pour développer son propre serveur. La politesse exige de demander l'autorisation de l'auteur par *email* avant de copier un élément graphique ou un programme.

Tous les navigateurs disposent d'une fonction qui permet de consulter le document source d'une page HTML. Certains permettent même qu'on choisisse une application particulière pour la consultation.

Ainsi sous Netscape Navigator on utilisera l'option `Document Source` du menu ***View***. On pourra en outre choisir d'utiliser un quelconque éditeur de texte plutôt que l'utilitaire de consultation intégré au navigateur qui ne permet ni d'enregistrer sur disque ni d'éditer le document chargé.

10.2.4 Les principales instructions (tags)

Sauts de ligne et de paragraphe, alignement, centrage

Ainsi que nous l'avons dit, les sauts de ligne ne sont pas pris en compte par les navigateurs.

Afin de restituer un saut de ligne au sein d'une mise page HTML, il faut donc faire appel à une instruction particulière. Les instructions <BR> et <P> permettent respectivement d'indiquer un saut de ligne et un saut de paragraphe. Elles se distinguent par la valeur de l'espace laissé entre deux lignes de texte.

En outre, la plupart des navigateurs ignorent les sauts de paragraphe consécutifs et ne tiennent compte que du premier. À l'origine, c'était également le cas pour les sauts de ligne. Les navigateurs Mosaic et Netscape ont un comportement différent à ce sujet, puisque Netscape Navigator tient compte des sauts de ligne multiples.

On notera que les instructions de saut ne sont pas suivies de leur inverse. Le saut de paragraphe est néanmoins assez souvent assimilé à une *marque* de paragraphe ; le texte correspondant est alors encadré par les instructions <P> et </P>. On utilise notamment cette possibilité pour changer l'alignement du bloc, en ajoutant à l'instruction le paramètre `ALIGN` qui prend pour valeur LEFT ou RIGHT.

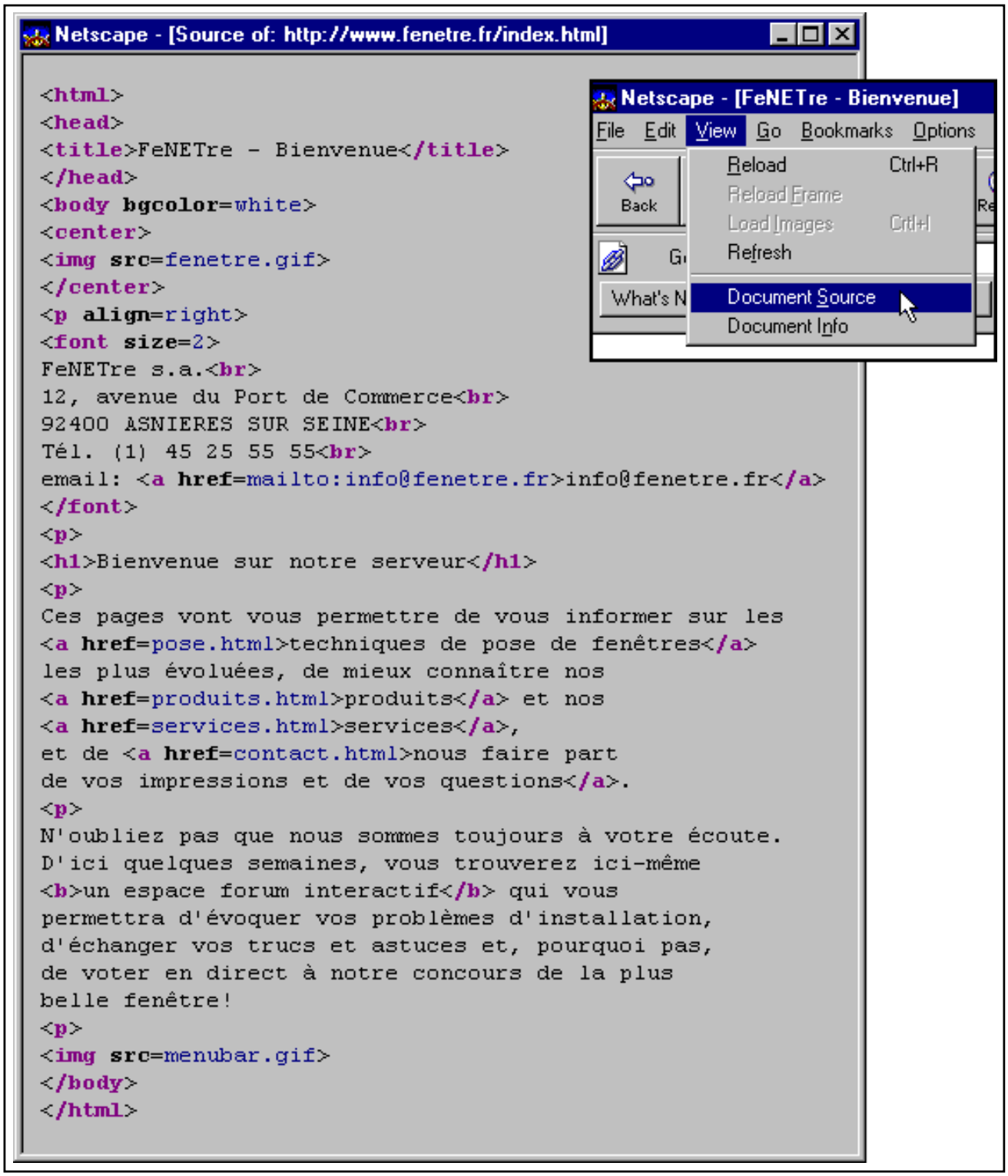

Figure 10.4 *Affichage du code source d'une page HTML*

```
FeNETre s.a.
<BR>
12, avenue du Port de Commerce
<BR>
92400 ASNIERES SUR SEINE
<BR>
Tél. (1) 45 25 55 55
<P>
<P>
<P>
<P>
Bienvenue sur notre serveur
```

Figure 10.5 *Format des sauts de ligne et de paragraphe en HTML*

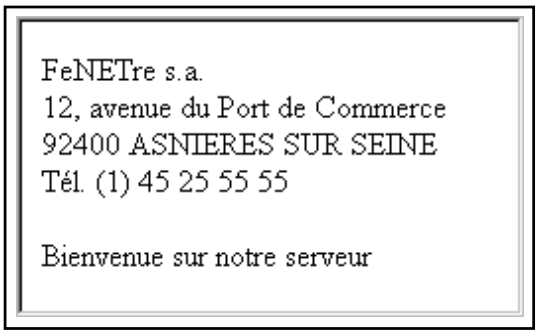

Figure 10.6 *Sauts de ligne et de paragraphe en HTML*

```
FeNETre s.a.
<P ALIGN=right>
12, avenue du Port de Commerce
<BR>
92400 ASNIERES SUR SEINE
<P>
Tél. (1) 45 25 55 55
```

Figure 10.7 *Saut de paragraphe et alignement à droite*

Figure 10.8 *Paragraphes alignés à droite en HTML*

Style	Instruction	Paramètres	Norme
Gras	`<B>...</B>`		
Italique	`<I>...</I>`		
Souligné	`<U>...</U>`		HTML 1. **Attention**, de nombreux navigateurs réservent le soulignement aux liens hypertexte et remplacent le style souligné par du gras ou des italiques ou n'en tiennent simplement pas compte.
Police à espacement fixe (télétype)	`<TT>...</TT>`		
Augmenter/diminuer la taille des caractères	`<FONT SIZE=+/-n>...</FONT>`	n=1... ; la taille par défaut est 3 et peut être changée par l'instruction <FONT SIZE=n> ci-dessous	HTML 2
Taille des caractères	`<FONT SIZE=n>...</FONT>`	n=1..7	HTML 2
Couleur	`<FONT COLOR=c>`	c=#RRVVBB ou nom de couleur	HTML 3

Figure 10.9 *Synoptique des styles de caractères HTML*

En revanche, on n'utilise pas l'instruction de paragraphe pour centrer un élément dans la page, mais plutôt l'instruction spéciale <CENTER> et son inverse. Elle permet de centrer n'importe quel objet : texte ou image, mais aussi les tableaux.

Styles de caractères

Les styles se rapportent à la forme ou la couleur des caractères. La graisse et les italiques font partie du premier niveau du standard. En revanche, les instructions utilisées pour changer la taille et la couleur des caractères relèvent respectivement des versions 2.0 et 3.0. À ce titre, elles ne sont pas reconnues par tous les navigateurs. De plus, tout comme l'ensemble des instructions HTML, elles ne sont données qu'à titre indicatif : ce qui signifie qu'il n'est pas possible de préjuger de l'aspect final de la page dans toutes les conditions. En particulier, les couleurs ne seront pas identiques d'un type de machine à un autre (selon la gestion de la palette des couleurs mise en œuvre par le navigateur), non plus que les tailles des polices (qui sont liées à la définition de l'écran et qui peuvent de toute façon être modifiées par l'utilisateur sur la plupart des navigateurs). On utilise donc les changements de taille de police pour mettre en évidence une hiérarchie dans le document (voir les niveaux de titre) ou pour signaler un point important, mais de préférence pas pour obtenir des effets précis de mise en page.

La figure 10.11 page 330 donne un exemple d'affichage sous Netscape 2.0.

Unités logiques

Les unités logiques ont un effet visuel similaire à ceux obtenus avec certains styles de caractères, mais elles définissent clairement la fonction logique du bloc encadré entre les instruc-

tions de début et de fin d'unité. Le résultat à l'affichage dépend du navigateur.

Mise en valeur	<STRONG>...</STRONG>
Source d'un programme ou d'un document HTML	<CODE>...</CODE>
Citation	<CITE>...</CITE>

```
Normal
<br>
<B>&lt;B&gt;Gras&lt;/B&gt;</B>
<br>
<I>&lt;I&gt;Italique&lt;/B&gt;</I>
<br>
<U>&lt;U&gt;Souligné&lt;/B&gt;</U>
<br>
<TT>&lt;TT&gt;Télétype&lt;/B&gt;</TT>
<p>
<FONT SIZE=-2>&lt;FONT SIZE=-2&gt;
(ou SIZE=1) &lt;/FONT&gt;</FONT>
<br>
<FONT SIZE=-1>&lt;FONT SIZE=-1&gt;
(ou SIZE=2) &lt;/FONT&gt;</FONT>
<br>
<FONT SIZE=+1>&lt;FONT SIZE=+1&gt;</FONT>
<br>
<FONT SIZE=+2>&lt;FONT SIZE=+2&gt;</FONT>
<br>
<FONT SIZE=+3>&lt;FONT SIZE=+3&gt;</FONT>
<br>
<FONT SIZE=+4>&lt;FONT SIZE=+4&gt;</FONT>
<br>
<FONT COLOR="gray">&lt;FONT COLOR="gray"&gt;Texte en gris&lt;/FONT&gt;</FONT>
<p>
<STRONG>&lt;STRONG&gt;Mise en valeur&lt;/STRONG&gt;</STRONG>
<br>
<CODE>&lt;CODE&gt;Source d'un programme
(HTML, C...)&lt;/CODE&gt;</CODE>
<br>
<CITE>&lt;CITE&gt;Citation&lt;/CITE&gt;</CITE>
```

Figure 10.10 *Instructions de formatage (styles) en HTML*

Niveaux de titres

Les niveaux de titres sont équivalents aux différents styles de titres appliqués dans les traitements de texte. Cependant la police ici ne change pas, seule sa taille et éventuellement sa graisse varient. Il y a six niveaux de titres, la taille des titres de niveau 4 est celle du texte normal. Notons qu'il ne s'agit pas uniquement d'un changement de taille et de graisse, puisque les instructions de titrage génèrent automatiquement un saut de paragraphe avant et après le titre. Pour changer de taille de police au sein d'un paragraphe, on utilise plutôt <FONT SIZE=???>.

Figure 10.11 *Utilisation des styles de caractères HTML*

Listes

Les trois types de listes disponibles sont les listes simples, les listes à points (ou puces) et les listes numérotées. Les listes peuvent être imbriquées, dans ce cas la marge à gauche du texte est d'autant plus importante que le niveau d'imbrication est élevé.

Certains navigateurs permettent de définir la forme des puces.

La liste simple se déclare avec l'instruction <DL>. Les instructions <DT> et <DD> servent respectivement à marquer une entrée dans la liste et la définition de cette entrée. Les définitions sont mises en retrait par rapport à l'ensemble du texte. On peut aussi utiliser ce type de

```
Texte normal
<H6>Titre 6</H6>
<H5>Titre 5</H5>
<H4>Titre 4</H4>
<H3>Titre 3</H3>
<H2>Titre 2</H2>
<H1>Titre 1</H1>
```

Figure 10.12 *Création de titres en HTML*

Figure 10.13 *Niveaux de titres HTML*

liste pour obtenir des effets de retrait à gauche sur des paragraphes (c'est d'ailleurs le seul moyen de le faire sans utiliser de tableaux).

```
<H2>Liste simple (deux niveaux)</H2>
<DL>
 <DT><B>nickname</B> (1) <I>n.</I>
 <DD>Surnom, sobriquet, diminutif
 <DT><B>nickname</B> (2) <I>v.tr.</I>
 <DD>
 <DL>
   <DT>1.
    <DD>Surnommer (qqun)
    <DT>2.
   <DD><I>(a)</I> Donner un sobriquet à qqun;
    <DD><I>(b)</I> appeler qqun par, de, son diminutif
 </DL>
</DL>
```

Figure 10.14 *Mise en place de listes simples en HTML*

Les listes à puces et numérotées se déclarent respectivement avec <UL> (pour *Unordered List*) et <OL> (pour *Ordered List*), et chaque entrée de la liste doit être précédée de <LI>. Le paramètre TYPE qui se place au sein des instructions UL ou LI (TYPE=disc, TYPE=circle, ou TYPE=square) permet de changer la forme des puces, respectivement pour toute la liste ou pour juste un des éléments.

On notera que des sauts de ligne et de paragraphe sont insérés automatiquement, respectivement entre les entrées de listes et après les listes elles-mêmes.

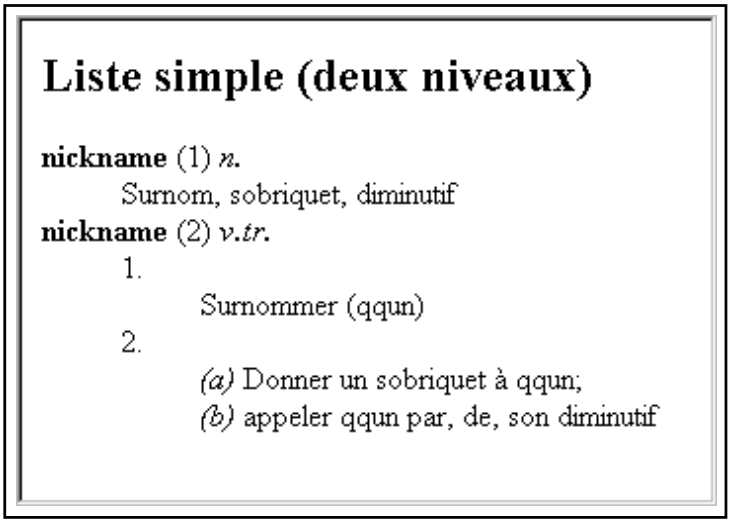

Figure 10.15 *Liste HTML simple*

```
<H2>Listes à puces et numérotées</H2>

<UL TYPE=square>
  <LI>Premier
  <LI>Second
  <LI TYPE=circle>Troisième
</UL>

<OL>
  <LI>Premier
  <LI>Second
  <LI>Troisième
</OL>
```

Figure 10.16 *Listes à puces ou numérotées en HTML*

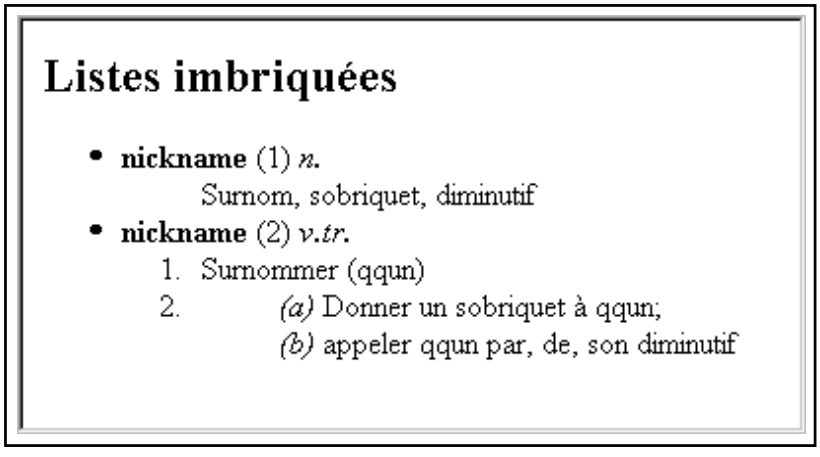

Figure 10.17 *Listes HTML*

```
<H2>Listes imbriquées</H2>

<UL>
 <LI><B>nickname</B> (1) <I>n.</I>
 <DL>
   <DD>Surnom, sobriquet, diminutif
 </DL>

 <LI><B>nickname</B> (2) <I>v.tr.</I>
 <OL>
    <LI>Surnommer (qqun)
    <LI>
    <DL>

     <DD><I>(a)</I> Donner un sobriquet à qqun;
      <DD><I>(b)</I> appeler qqun par, de, son diminutif
   </DL>
 </OL>

</UL>
```

Figure 10.18 *Imbrication de listes HTML*

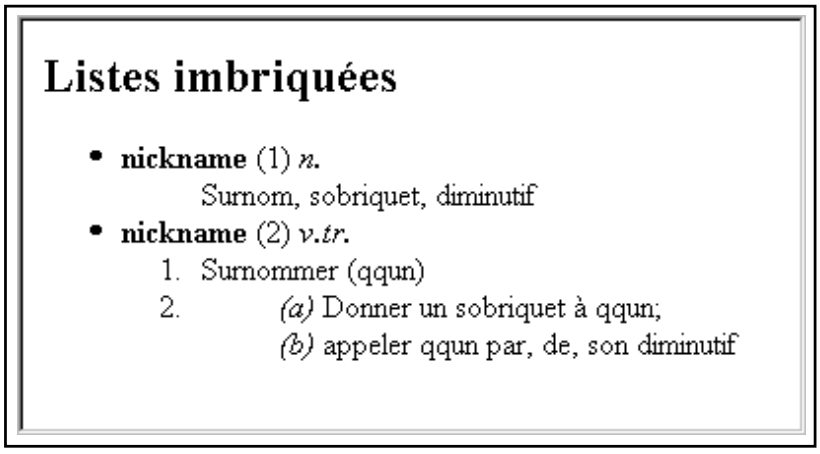

Figure 10.19 *Listes imbriquées*

Tableaux

L'usage des tableaux peut se révéler assez compliqué, surtout lorsqu'ils sont utilisés pour forcer une mise en page plus stricte qu'avec les instructions HTML habituelles. Cependant, lorsqu'on n'y fait appel que pour formater des données, leur maniement reste abordable. Malheureusement, il existe encore peu d'éditeurs de pages HTML qui permettent de concevoir des tableaux complexes de manière interactive et entièrement à la souris.

La structure générale d'un tableau HTML (<TABLE>), disponible dans les versions 2.0 et suivantes, s'articule autour d'une division en lignes (<TR> pour *table row*), elles-mêmes divisées en cellules (<TD> pour *table data*), comme le montre la figure 10.20 page suivante.

Cependant, cette structure relativement classique présente certaines particularités par rapport à celle des tableaux proposés par les logiciels de traitement de texte habituels.

```
<TABLE BORDER=1>
<TR><TD>Cellule 1</TD><TD>Cellule 2</TD></TR>
<TR><TD>Cellule 3</TD><TD>Cellule 4</TD></TR>
<TR><TD>Cellule 5</TD><TD>Cellule 6</TD></TR>
</TABLE>
```

Figure 10.20 *Instructions génériques des tableaux HTML*

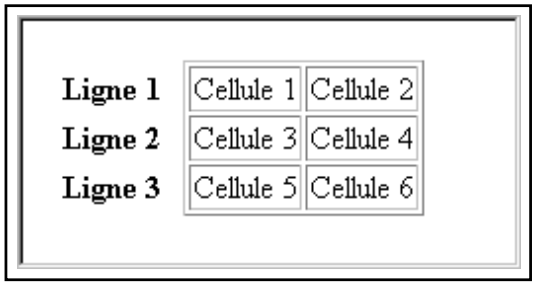

Figure 10.21 *Aspect général d'un tableau HTML*

Le paramètre BORDER permet de spécifier l'épaisseur en pixels du cadre qui entoure chacune des cellules. Certains navigateurs affichent ce cadre avec un effet de relief. Sa couleur est définie par la couleur du texte et celle du fond, d'une manière qui dépend du système.

Au sein d'un tableau, la largeur des cellules d'une même colonne est nécessairement constante. Il est cependant possible de fusionner plusieurs cellules, que ce soit en hauteur ou en largeur, grâce aux paramètres ROWSPAN (littéralement « étendre sur (plusieurs) lignes ») et COLSPAN.

```
<TABLE BORDER=1>
<TR><TD>Cellule 1</TD><TD>Cellule 2</TD></TR>
<TR><TD ROWSPAN=2>Cellule 3+5</TD><TD>Cellule 4</TD></TR>
<TR><TD>Cellule 6</TD></TR>
<TR><TD COLSPAN=2>Cellules 7+8</TD></TR>
</TABLE>
```

Figure 10.22 *Cellules étendues dans un tableau HTML*

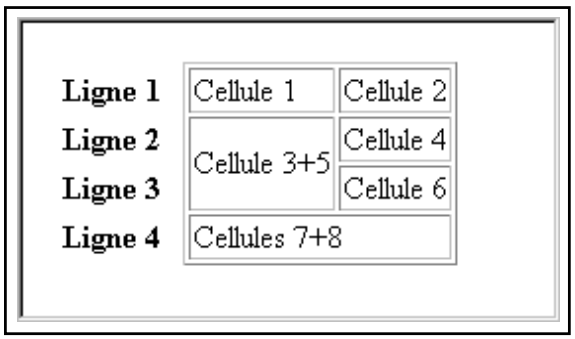

Figure 10.23 *Cellules de plusieurs lignes et/ou plusieurs colonnes*

Il est possible de définir les dimensions du tableau, soit en pixels soit en pourcentage des

dimensions de la fenêtre d'affichage. Les dimensions des cellules elles-mêmes peuvent être fixées, mais ces dimensions ne sont pas toujours respectées par les navigateurs. De plus, si la largeur du contenu d'une cellule est supérieure à celle de la cellule, le texte ou les images risquent de déborder.

```
<TABLE BORDER=1>
<TD WIDTH=30>
ANTICONSTITUTIONNELLEMENT
</TD>
</TABLE>
```

Figure 10.24 *Débordement du texte dans une cellule en HTML*

Figure 10.25 *Débordement du contenu d'une cellule dans un tableau HTML*

Pour être raisonnablement sûr que les dimensions demandées seront utilisées, il est préférable de contraindre l'ensemble du tableau, c'est-à-dire de préciser à la fois les dimensions des cellules et celles du tableau lui-même (en s'assurant que ces informations soient cohérentes).

On utilise de plus en plus souvent les tableaux pour contraindre l'aspect d'un document, en particulier lorsqu'on souhaite articuler entre eux les divers éléments graphiques d'une page. Il s'agit là d'une tentative (malheureusement souvent maladroite) de retrouver sur le Web les fonctions offertes par les logiciels de PAO. Signalons que, même fortement contrainte, la structure de la page reste intimement liée à ce que tel ou tel navigateur est capable d'afficher. Ce genre de procédé doit donc être utilisé avec prudence, après s'être assuré que le résultat restera lisible même si le navigateur ne formate pas correctement le document.

L'alignement des éléments au sein des cellules d'un tableau peut se faire grâce aux instructions habituelles, mais on pourra leur préférer les paramètres `ALIGN` (`LEFT`, `CENTER`, `RIGHT`) et `VALIGN` (`TOP`, `CENTER`, `BOTTOM`) qui s'appliquent à l'ensemble de la cellule. L'alignement par défaut est à gauche et centré en hauteur :

```
ALIGN=LEFT VALIGN=CENTER
```

Enfin, deux autres paramètres du tableau permettent de définir l'espace entre les cellules (`CELLSPACING`) et les marges aménagées autour du contenu des cellules (`CELLSPACING`). Comme toutes les autres dimensions, ces paramètres sont indiqués en pixels ou en pourcentages (de la dimension du tableau).

```
<TABLE WIDTH=450 HEIGHT=210 BORDER=1 CELLPADDING=0 CELLSPACING=0>
<TR>
 <TD WIDTH=150 HEIGHT=70 VALIGN=TOP>
   <FONT SIZE=1>ALIGN=LEFT (défaut)<BR>VALIGN=TOP</FONT>
 </TD>
 <TD WIDTH=150 HEIGHT=70 ALIGN=CENTER VALIGN=TOP>
   <FONT SIZE=1>ALIGN=CENTER<BR>VALIGN=TOP</FONT>
 </TD>
 <TD WIDTH=150 HEIGHT=70 ALIGN=RIGHT VALIGN=TOP>
   <FONT SIZE=1>ALIGN=RIGHT<BR>VALIGN=TOP</FONT>
 </TD>
</TR>
<TR>
  <TD WIDTH=150 HEIGHT=70>
   <FONT SIZE=1>ALIGN=LEFT (défaut)<BR>VALIGN=CENTER (défaut)</FONT>
 </TD>
 <TD WIDTH=150 HEIGHT=70 ALIGN=CENTER>
   <FONT SIZE=1>ALIGN=CENTER<BR>VALIGN=CENTER (défaut)</FONT>
 </TD>
 <TD WIDTH=150 HEIGHT=70 ALIGN=RIGHT>
   <FONT SIZE=1>ALIGN=RIGHT<BR>VALIGN=CENTER (défaut)</FONT>
 </TD>
</TR>
<TR>
 <TD WIDTH=150 HEIGHT=70 VALIGN=BOTTOM>
    <FONT SIZE=1>ALIGN=LEFT (défaut)<BR>VALIGN=BOTTOM</FONT>
 </TD>
 <TD WIDTH=150 HEIGHT=70 ALIGN=CENTER VALIGN=BOTTOM>
   <FONT SIZE=1>ALIGN=CENTER<BR>VALIGN=BOTTOM</FONT>
 </TD>
 <TD WIDTH=150 HEIGHT=70 ALIGN=RIGHT VALIGN=BOTTOM>
   <FONT SIZE=1>ALIGN=RIGHT<BR>VALIGN=BOTTOM</FONT>
 </TD>
</TR>
</TABLE>
```

Figure 10.26 *Utilisation de dimensions numériques dans un tableau HTML*

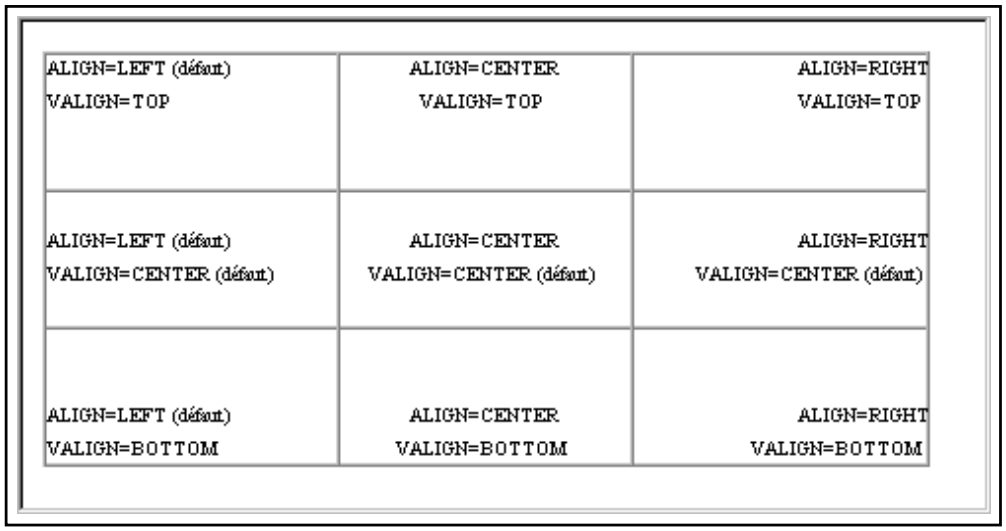

Figure 10.27 *Contraintes d'alignement, d'espacement et de dimensions dans un tableau HTML*

Images

L'insertion d'images au sein d'une page HTML s'effectue également à l'aide d'une instruction particulière. Bien entendu, le document source ne contient pas les données de l'image mais uniquement une référence à son URL. C'est le rôle du navigateur de télécharger l'image lorsqu'il rencontre l'instruction correspondante, et à condition que l'utilisateur n'ait pas demandé à consulter uniquement le texte.

L'instruction <IMG> permet de spécifier l'URL d'une image à insérer dans le document. Elle accepte un certain nombre de paramètres, dont un au moins est obligatoire puisqu'il s'agit de l'URL lui-même, indiqué sous la forme SRC="URL". Une des grandes force du Web provient de cette capacité à mêler au sein d'une même page des images pouvant provenir de n'importe quel serveur dans le monde.

```
<IMG SRC="logo.gif">
<P>
<CENTER>
<IMG SRC="http://www.intuisys.fr/fr/im/intuisys.gif">
</CENTER>
```

Figure 10.28 *Mise en place d'une image à partir de son URL*

Figure 10.29 *Insertion d'une image distante au sein d'une page HTML*

Pour formater le document avant de l'afficher, les navigateurs ont besoin de connaître les dimensions des images (les dernières versions de Netscape Navigator et Microsoft Explorer affichent le texte une première fois en laissant un espace de taille fixe pour les images, puis affichent la version définitive lorsque la taille des images est connue). Par conséquent, si cette dernière n'est pas indiquée en paramètre (`WIDTH`, `HEIGHT` en pixels ou pourcentage des dimensions de la fenêtre d'affichage) dans l'instruction <IMG>, le formatage ne sera possible que lorsque les images auront été chargées, ce qui retarde généralement le moment ou le navigateur affiche le texte. Il est donc préférable de préciser systématiquement les dimensions des images : le texte apparaît alors en premier, la lecture des pages est facilitée. Les paramètres de dimension d'image sont reconnus par la plupart des navigateurs au standard HTML 2.0.

On remarquera qu'il est possible de spécifier une largeur et une hauteur différentes de celles de l'image originale. Dans ce cas, l'image sera redimensionnée avant d'être affichée. Il est

ainsi possible de jouer sur la taille apparente des images sans toutefois que le navigateur télécharge plusieurs fois le fichier. Cependant, en raison des traitements appliqués pour réduire le nombre de couleurs utilisées, les images au format GIF supportent parfois mal le redimensionnement et laissent apparaître des phénomènes de tramage (figure 10.30).

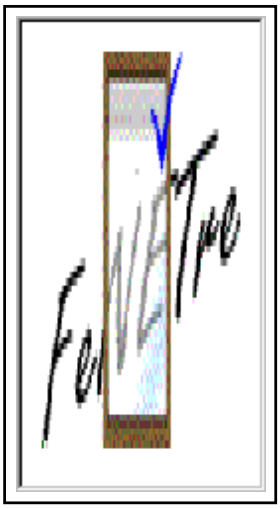

Figure 10.30 *Apparition d'une trame dans une image basse définition redimensionnée*

Lorsqu'un lien hypertexte est associé à une image, l'image apparaît encadrée d'un filet de couleur (qui remplace en quelque sorte le soulignement appliqué au texte). Il est parfois souhaitable de supprimer ce filet, ou à l'inverse d'augmenter son épaisseur. La plupart des navigateurs reconnaissent le paramètre BORDER qui précise l'épaisseur du filet en pixels.

Enfin Netscape a défini le paramètre `ALIGN` qui permet entre autres de spécifier la position verticale de l'image par rapport au texte.

```
<IMG SRC=image.gif> Normal (le bas de l'image est posé sur la ligne de base)
<P>
<IMG ALIGN=CENTER SRC=image.gif> "Centré" (le centre de l'image est aligné sur
la ligne de base)
<P>
<IMG ALIGN=TOP SRC=image.gif> Haut (le haut de l'image est aligné
avec le haut du texte)
```

Figure 10.31 *Ajustement d'une image par rapport à la ligne de base du texte HTML*

Word Flow

Certains navigateurs sont capables de formater le texte autour des images, tout comme le ferait un logiciel de PAO. C'est une autre utilisation du paramètre ALIGN de l'instruction <IMG>. ALIGN=LEFT indique que l'image doit être placée à gauche du texte, tandis que ALIGN=RIGHT la renvoie à droite. Si plusieurs images sont mises à droite, la première à être déclarée sera à l'extrémité.

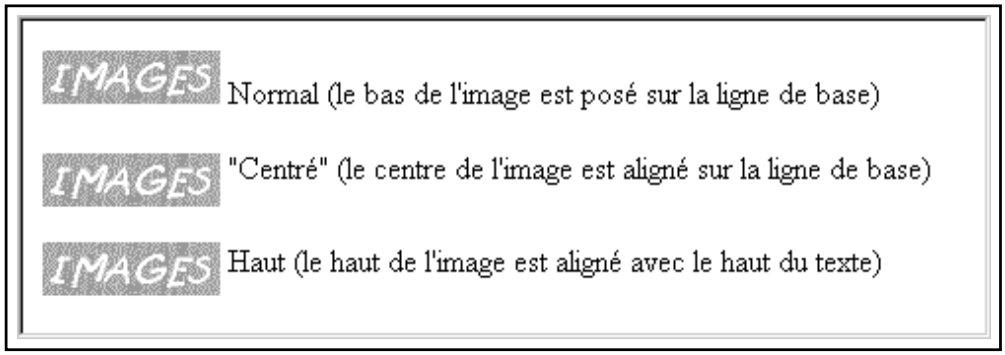

Figure 10.32 *Position des images par rapport au texte HTML*

Les paramètres HSPACE et VSPACE déterminent la taille des marges à laisser entre l'image et le texte, respectivement horizontalement et verticalement.

```
<IMG ALIGN=RIGHT SRC=logo.gif>
<IMG ALIGN=LEFT HSPACE=6 VSPACE=4 SRC=papillon.gif>
<H2>
Le texte épouse la forme des images placées à droite et à gauche jusqu'à ce qu'il
en atteigne le bas; il occupe alors de nouveau toute la ligne.
</H2>
```

Figure 10.33 *Ajustement du texte HTML autour d'une image (wordflow)*

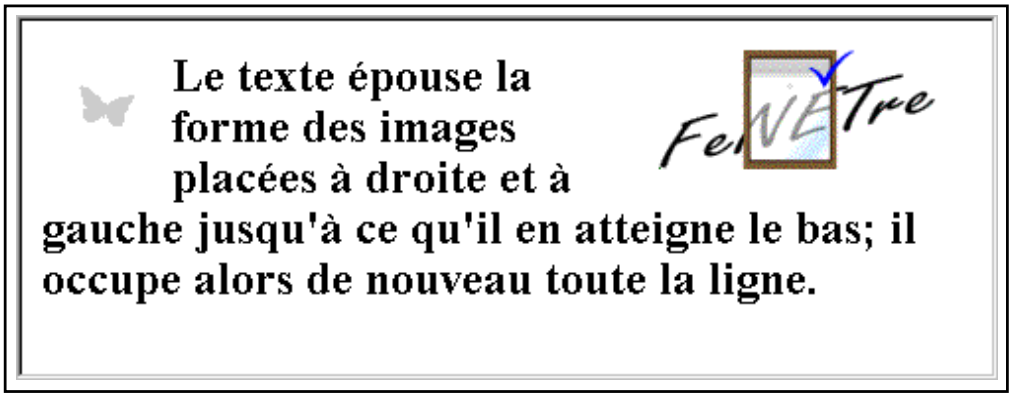

Figure 10.34 *Texte HTML entourant une image*

L'instruction <BR> a été munie d'un paramètre complémentaire afin qu'il soit possible de laisser autant d'espace que nécessaire pour dépasser le bas de l'image. Ainsi :

<BR CLEAR=LEFT>

positionnera le texte à suivre sous l'image située à gauche (voir l'exemple final).

<BR CLEAR=ALL> garantit que la largeur est libre de toute image.

Traits de séparation

L'instruction <HR> permet de faire apparaître un trait centré, en général avec un effet de relief et occupant toute la largeur de la page (ou de la cellule du tableau). Le paramètre `SIZE` permet de changer son épaisseur qui est par défaut de deux pixels. Le paramètre `WIDTH` indique la largeur du trait, en pixels ou en pourcentage de la largeur de l'écran. Enfin le paramètre `NOSHADE` supprime l'effet de relief. La couleur du trait est la même que celle des cadres pour les tableaux.

```
<HR>
<HR NOSHADE WIDTH=40%>
<HR SIZE=4 WIDTH=20%>
<HR NOSHADE SIZE=4>
```

Figure 10.35 *Insertion de traits horizontaux*

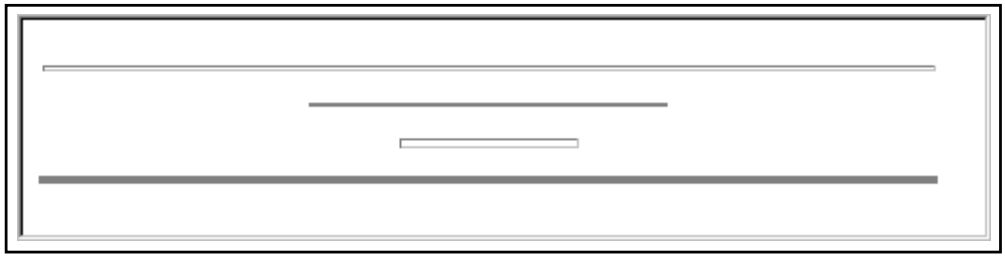

Figure 10.36 *Traits de séparation en HTML*

10.2.5 Liens hypertexte et ancres

La richesse des documents consultables sur le Web réside bien évidemment dans la possibilité de les enrichir de références croisées sous forme de liens hypertexte. Ces liens peuvent se placer sur des images ou sur des fragments de texte, grâce à une instruction HTML.

On distingue les liens permettant de sauter d'un document à un autre des liens désignant des parties spécifiques d'un même document. À cet effet, le langage HTML enrichit la notation des URL définie par le protocole http en y ajoutant la notion d'ancre (*anchor*). Ces ancres sont simplement des noms qu'on associe à un point particulier du document grâce à l'instruction <A NAME="nom de l'ancre">. Elles sont mentionnées au sein des URL en faisant suivre le nom du document du signe # et du nom de l'ancre.

On associe par ailleurs une destination à un lien sur du texte ou une image grâce à l'instruction <A HREF="URL">. Si l'URL en question est une ancre au sein du même document, le nom du document peut être omis. Ainsi dans la présentation générale des services de la société FeNETre, on trouve trois parties principales : pose, service après-vente et conseil. Les concepteurs du site ont choisi de conserver ces trois parties sur une même page afin d'éviter la prolifération des pages presque vides. Cependant, ils ont souhaité que chacune de ces parties

soit accessible directement depuis un menu situé en haut de page, afin que l'utilisateur n'ait pas à utiliser l'ascenseur de la fenêtre pour rechercher l'information qui l'intéresse.

Chaque partie a été munie d'une ancre, respectivement *pose*, *sav* et *conseil*. Un menu composé de trois liens hypertexte en haut de page permet de descendre directement à la partie souhaitée. Techniquement, il n'est pas nécessaire de *fermer* l'instruction <A NAME=???>, puisqu'elle marque un point précis du document et n'entraîne aucun effet visuel. Cependant, pour conserver la cohérence du document et simplifier la tâche d'éventuels programmes de mise à jour automatique, il est souhaitable de ne pas omettre l'instruction </A>, immédiatement après la déclaration de l'ancre, ou plus loin dans le texte (par exemple après le titre du paragraphe).

En cliquant sur un des éléments du menu, on positionne automatiquement le texte souhaité dans la fenêtre active, sans avoir à descendre à l'aide du clavier ou de la souris, comme le montrent les figures 10.38 page suivante et 10.39 page 343.

La page contient également un lien hypertexte vers une autre page du serveur. L'URL indiqué dans le source HTML est relatif au serveur et au chemin d'accès courant : c'est le navigateur qui le complétera afin de fabriquer une requête cohérente.

Ici, l'URL de la page courante est :

```
http://www.fenetre.fr/presente/services.html
```

L'URL de la page `livraison` sera donc :

```
http://www.fenetre.fr/presente/livraison.html
```

Elle apparaît dans la barre d'état de la fenêtre du navigateur lorsqu'on place la souris au dessus du lien. Bien entendu, on peut utiliser la notation `../` pour remonter d'un niveau dans l'arborescence ; cependant, il ne sera pas possible de remonter plus haut sur le disque que jusqu'au répertoire `DocumentRoot` lui-même.

Enfin pour indiquer un lien vers un autre serveur, il est nécessaire de spécifier un URL absolu. L'erreur la plus courante consiste à omettre le protocole (figures 10.41 page 343 et 10.42 page 343). Dans ce cas, le navigateur considérera comme première partie du chemin d'accès ce qui est en fait le nom de la machine. Ici les liens encadrent des images ; la syntaxe est exactement la même que pour du texte. On peut également mélanger les deux. Pour supprimer l'encadrement autour d'une image ou au contraire augmenter sa taille, il faut utiliser le paramètre `BORDER` de l'instruction <IMG>.

On veillera à ne pas commettre d'erreur de mise en page : si l'instruction qui ferme le lien (</A>) est séparée du texte ou de l'image par un espace (ou un saut de ligne qui sera traduit par un espace à l'affichage), certains navigateurs souligneront cet espace – ce qui ne manquera pas de nuire à l'esthétique de la page.

```
<H1>FeNETre S.A.</H1>
<H2>Nos services</H2>

<UL>
<LI>\<A HREF=#pose>La <A HREF=livraison.html>livraison</A> et la pose sous 48H</A>
<LI><A HREF=#sav>Nos garanties de service après-vente</A>
<LI><A HREF=#conseil>Notre équipe de conseillers à votre disposition</A>
</UL>
Chez FeNETre, le service n'est pas un vain mot.
<P>


Depuis plus de 10 ans [...]


<P>
<A NAME=pose></A>
<H3>La livraison et la pose en moins de 48H, montre en main!</H3>
Si vous avez décidé [...]


<A NAME=sav>
<H3>L'après-vente sans soucis</H3>
</A>


Toutes nos fenêtres [...]


<A NAME=conseil>
<H3>Une question, un problème délicat ? Demandez-nous conseil !</H3>
</A>

Nos collaborateurs [...]
```

Figure 10.37 *Insertion d'ancres et de liens HTML*

Figure 10.38 *Ancres dans une page HTML*

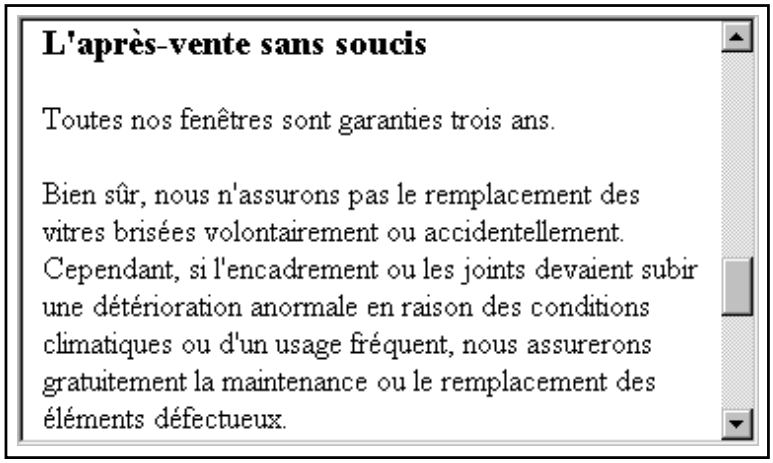

Figure 10.39 *Positionnement au sein d'une page HTML*

Figure 10.40 *Lien hypertexte dans une page HTML*

```
Essayez de cliquer
<A HREF="www.fenetre.fr/presente/livraison.html">
ici</A> pour aller à la page des livraisons.
<P>
<A HREF="http://www.fenetre.fr/presente/livraison.html">
<IMG SRC="main.gif"> Ce lien</A> fonctionnera sans doute mieux.
```

Figure 10.41 *Erreur de méthode d'accès sur un lien hypertexte*

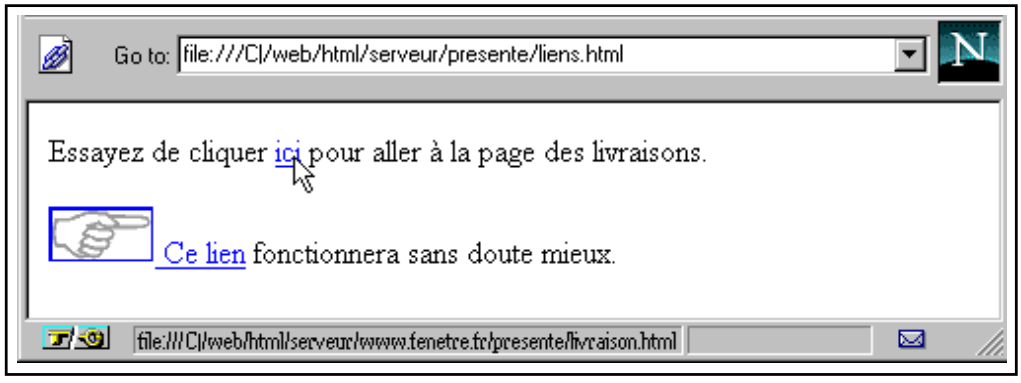

Figure 10.42 *Cadre par défaut autour d'une image utilisée comme hyperlien*

```
Essayez de cliquer
<A HREF="file:///C|/web/html/serveur/www.fenetre.fr/presente/livraison.html">
ici</A> pour aller à la page des livraisons.
<P>
<A HREF="http://www.fenetre.fr/presente/livraison.html">
<IMG BORDER=0 SRC="main.gif"> Ce lien</A> fonctionnera sans doute mieux.
```

Figure 10.43 *Utilisation de BORDER pour supprimer le cadre autour d'une image*

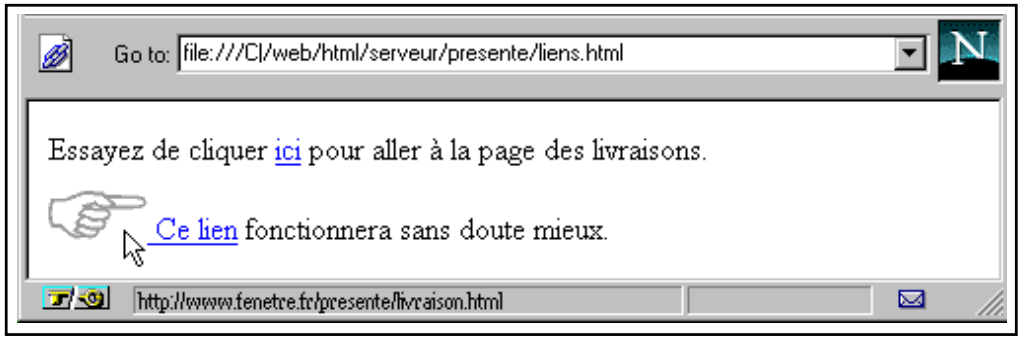

Figure 10.44 *Suppression du cadre autour d'une image utilisée comme hyperlien*

```
Attention à ne pas laisser d'espace
ou de saut de ligne
entre le <A HREF="">texte des liens </A>
et \&lt;/A\&gt;
<BR>
C'est d'autant plus important
pour les <A HREF="">
<IMG BORDER=3 SRC="image.gif">
</A>
car le soulignement indésirable
est particulièrement visible.
```

Figure 10.45 *Mauvaise mise en place des liens ou ancres*

Figure 10.46 *Débordement du trait de soulignement des liens hypertexte*

TEXT	La couleur du texte, sous la forme #RRVVBB (rouge-vert-bleu en hexadécimal) ou en précisant un nom de couleur.
LINK	La couleur des liens qui pointent vers un document qu'on n'a pas encore vu.
VLINK	La couleur des liens qui pointent vers un document qu'on a déjà vu.
ALINK	La couleur des liens pendant qu'on clique dessus (avant de relâcher le bouton de la souris).
BGCOLOR	La couleur du fond. Elle sert parfois (décors 3D sous Netscape) à déterminer la couleur des encadrements de tableaux ainsi que les traits horizontaux (<HR>). Il est donc préférable d'en tenir compte même si on utilise une image de fond.
BACKGROUND	L'URL d'une image de fond. Le cas échéant, l'image sera répétée horizontalement et verticalement pour couvrir toute la surface de la fenêtre d'affichage.

Figure 10.47 *Paramètres de couleur pour les pages HTML*

Liens vers des adresses email ou ftp

Les liens mis en place font référence à un URL : à ce titre, il est tout à fait possible d'indiquer une autre méthode (un autre protocole) que `http://`. En particulier, on utilise souvent les liens hypertexte pour indiquer un lien vers une adresse *email* ou un site FTP. Si le navigateur comprend ces protocoles, il agira en conséquence, en lançant une interface de courrier électronique ou en téléchargeant le fichier. Dans ses versions 2.0 et suivantes, Netscape Navigator intègre une application permettant de lire et d'envoyer des courriers électroniques. Ainsi, lorsque l'utilisateur sélectionne un lien de la forme <A HREF=mailto:contact@fenetre.fr>, le navigateur lance automatiquement l'outil correspondant en lui transmettant l'adresse à laquelle envoyer un message.

10.2.6 Couleur ou image de fond de page et couleur du texte

L'instruction <BODY> qui sert à indiquer le début du corps du document accepte un certain nombre de paramètres qui permettent de fixer la couleur par défaut du texte et des liens hypertexte, ainsi que la couleur du fond et éventuellement l'image de fond. Ces indications ne seront peut-être pas respectées, notamment si l'utilisateur a configuré son navigateur en lui imposant les couleurs de son choix.

Les couleurs sont indiquées soit par leur nom (*red*, *green*...), soit par leur dosage en rouge, vert et bleu.

Il n'y a pas de standard pour la couleur de fond par défaut. Certains navigateurs s'en tiennent encore au gris souris utilisé par les premières versions de Mosaïc, d'autres tel Netscape ont adopté le blanc. Pour être raisonnablement sûr que la couleur de fond sera bien le blanc sur la majorité des écrans (sauf choix contraire de l'utilisateur ou modèle de navigateur ne reconnaissant pas le paramètre `BGCOLOR`) il est indispensable de la préciser systématiquement.

```
<BODY BGCOLOR=\#FFFFFF TEXT=\#666666 LINK=\#CCCCCC VLINK=black ALINK=red>
<B>
<FONT SIZE=+2>
Texte en gris moyen (\#666666)
<P>
<A HREF="">Lien en gris clair (\#CCCCCC)</A>
<P>
<A HREF="colors.html">Lien déjà visité en noir (black)</A>
<P>
<A HREF="colors.html">Lien sélectionné en rouge (red)</A>
</FONT>
</B>
```

Figure 10.48 *Choix d'un jeu de couleurs pour le texte et les hyperliens*

Figure 10.49 *Couleurs du texte et des différents types de liens*

10.2.7 Exemple de page complète

La figure 10.51 page 348 présente un exemple de page HTML complète, intégrant une image de fond et des menus.

10.2.8 Frames

Les *frames*, apparues récemment, permettent de découper la fenêtre d'affichage du navigateur en plusieurs fenêtres adjacentes, dans lesquelles on peut afficher autant de pages différentes.

Leur utilisation principale est double : conserver en permanence à l'écran un menu permettant de naviguer dans le site, et dispenser des informations complémentaires à celles déjà présentes sur la page principale. Elles sont également particulièrement utiles pour la mise en place de formulaires complexes à plusieurs niveaux et qu'on doit pouvoir modifier à tout instant.

Les frames sont avec les tableaux l'autre moyen de contraindre les dimensions des différentes parties d'une mise en page HTML. Cependant, et contrairement aux tableaux, il s'agit véritablement de fenêtres au sens de l'interface graphique : à ce titre, elles sont affublées des

```
<HTML>

<HEAD>
<TITLE>FeNETre - Bienvenue</TITLE>
</HEAD>

<BODY BGCOLOR=WHITE BACKGROUND="fond.gif">

<CENTER>
  <IMG SRC=fenetre.gif>
</CENTER>

<P ALIGN=right>
<FONT SIZE=2>
FeNETre s.a.<BR>
12, avenue du Port de Commerce<BR>
92400 ASNIERES SUR SEINE<BR>
Tél. (1) 45 25 55 55<BR>
email: <A HREF=mailto:info@fenetre.fr>info@fenetre.fr</A>
</FONT>
<P>

<H1>Bienvenue sur notre serveur !</H1>

<IMG ALIGN=LEFT SRC="exclusif.gif">
<B>
Nous mettons aussi des fenêtres dans notre serveur!!! Munissez-vous d'un
navigateur capable d'afficher des <I>frames</I> (Navigator 2.x ou Explorer 3.x)
et visitez la <A HREF=frames.html>version véranda</A> de nos pages!
</B>
<BR CLEAR=ALL>
<BR>
Ces pages vont vous permettre de vous informer sur les
<A HREF=pose.html>techniques de pose de fenêtres</A> les plus évoluées, de
mieux connaître nos <A HREF=produits.html>produits</A> et nos
<A HREF=services.html>services</A>, et de <A HREF=contact.html>nous faire part
de vos impressions et de vos questions</A>.
<P>
N'oubliez pas que nous sommes toujours à votre écoute. D'ici quelques semaines,
vous trouverez ici-même <B>un espace forum interactif</B> qui vous permettra
d'évoquer vos problèmes d'installation, d'échanger vos trucs et astuces et,
pourquoi pas, de voter en direct à notre concours de la plus belle fenêtre!
<P>

<CENTER>
  <A TARGET=page HREF=menu.map>
 <IMG BORDER=0 ISMAP USEMAP=\#menu SRC=menubar.gif WIDTH=562 HEIGHT=20></A>
</CENTER>

<MAP NAME=menu>
  <AREA COORDS=0,0,60,20 HREF=societe.html>
  <AREA COORDS=90,0,160,20 HREF=produits.html>
  <AREA COORDS=190,0,245,20 HREF=services.html>
  <AREA COORDS=280,0,345,20 HREF=contacts.html>
  <AREA COORDS=385,0,430,20 HREF=forum.html>
  <AREA COORDS=470,0,562,20 HREF=partenaires.html>
</MAP>

</BODY>

</HTML>
```

Figure 10.50 *Source d'une page HTML complète*

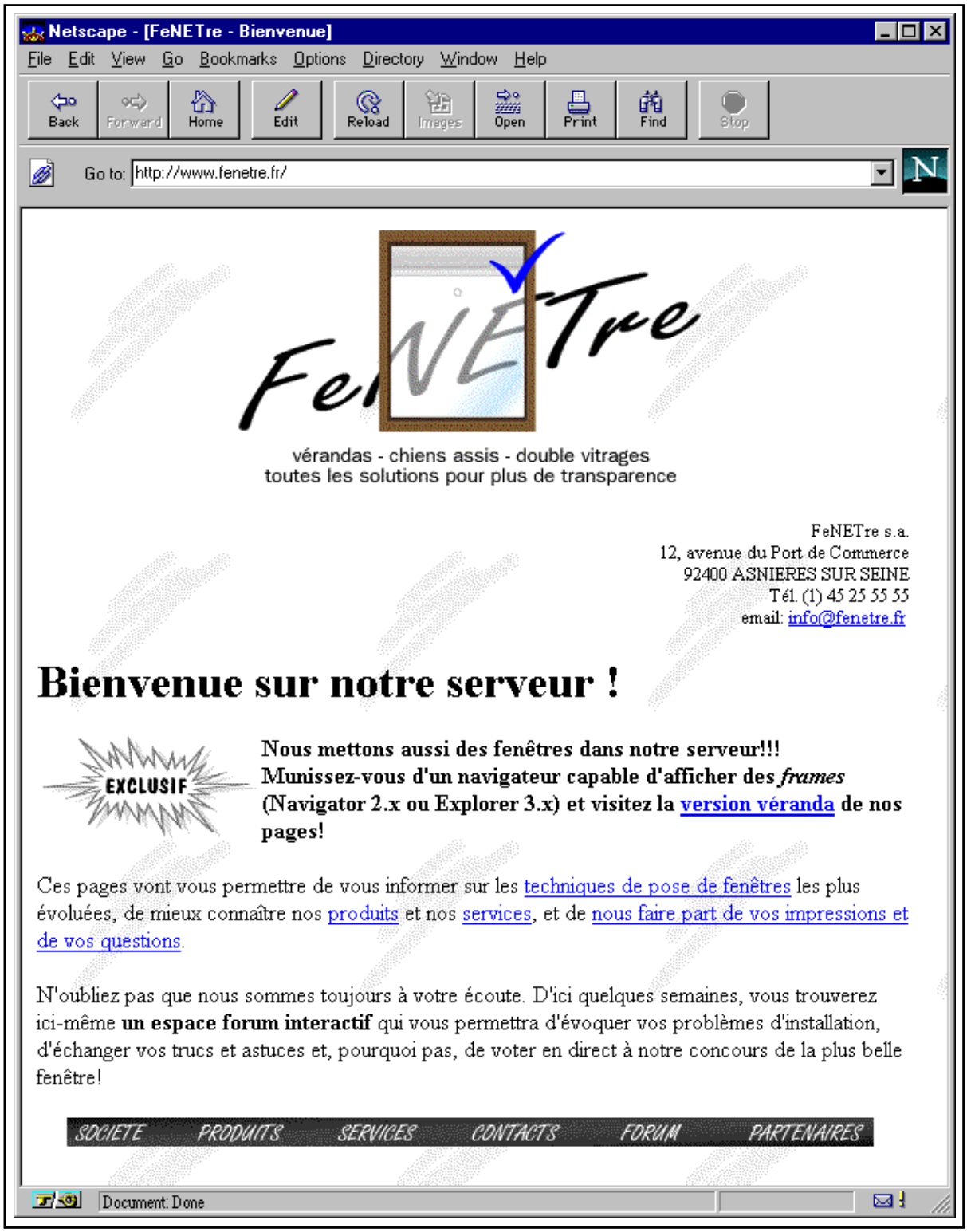

Figure 10.51 *Exemple de page d'accueil complète*

mêmes cadres et décors 2D ou 3D que les autres fenêtres (à l'exception des barres de titre, d'outils et d'état bien entendu).

Cela signifie que, d'une part, leur apparence pourra varier d'un système à l'autre, d'autre part que les dimensions exactes de chacune des fenêtres ne sont pas connues dans la mesure où les cadres empiètent sur l'espace d'affichage. On s'interdira de préférence de les utiliser pour générer des effets de présentation, les réservant à des situations où leur emploi se justifie véritablement.

La mise au point de pages évoluées comportant un système de frames liées entre elles par des références croisées n'est pas du ressort de cet ouvrage ; nous nous contenterons de montrer un exemple très simple d'une page surmontée d'un menu fixe.

On notera l'utilisation des cartes cliquables ou *Image Map* (décrites en 10.3.3 page 360).

frames.html

```
<TITLE>FeNETre</TITLE>
<FRAMESET ROWS=40,*>
 <FRAME NAME=menu SRC=menu.html MARGINHEIGHT=4 MARGINWIDTH=1 SCROLLING=no NORESIZE>
 <FRAME NAME=page SRC=index.html MARGINHEIGHT=7 MARGINWIDTH=7 SCROLLING=auto NORESIZE>
</FRAMESET>
```

menu.html

```
<HTML>
<BODY BGCOLOR=white>
<CENTER>
  <A TARGET=page HREF=menu.map>
 <IMG BORDER=0 ISMAP USEMAP=\#menu SRC=menubar.gif WIDTH=562 HEIGHT=20></A>
</CENTER>

<MAP NAME=menu>
  <AREA COORDS=0,0,60,20 HREF=societe.html>
  <AREA COORDS=90,0,160,20 HREF=produits.html>
  <AREA COORDS=190,0,245,20 HREF=services.html>
  <AREA COORDS=280,0,345,20 HREF=contacts.html>
  <AREA COORDS=385,0,430,20 HREF=forum.html>
  <AREA COORDS=470,0,562,20 HREF=partenaires.html>
</MAP>

</BODY>
</HTML>
```

Figure 10.52 *Ajout de frames à une page HTML*

10.2.9 Formulaires

Le principe des formulaires HTML consiste à proposer au sein d'une page un ensemble de zones de saisie, et à expédier le résultat vers un programme capable de traiter les données ainsi entrées par l'utilisateur.

Figure 10.53 *Page d'accueil en version "frames"*

Nous verrons que les valeurs des champs peuvent être passées au programme en paramètres, par l'intermédiaire de variables d'environnement, ou encore par l'entrée standard.

Le format des échanges, et en particulier l'encodage auquel sont soumises les données, est formalisé par la norme CGI (*Common Gateway Interface*) qui est utilisée pour interfacer des scripts et applications divers avec les serveurs HTTP.

Nous allons pour le moment insérer quelques champs de saisie au sein d'un formulaire ; l'interface CGI et son utilisation seront étudiées section 10.3.1 page 354.

La structure d'un formulaire est assez constante, puisqu'il se compose nécessairement d'un ou plusieurs champs de texte, boutons ou cases à cocher, et invariablement d'un bouton de validation destiné à lancer le transfert des données et leur prise en compte par le programme de gestion situé sur le serveur.

Pour placer un formulaire au sein d'une page, il convient de le délimiter par l'instruction <FORM> et son inverse </FORM>.

Ces instructions peuvent être placées à n'importe quel endroit de la page, du moment qu'elles encadrent l'ensemble des instructions du formulaire lui-même, c'est-à-dire celles qui déterminent l'affichage des champs de texte, des boutons ou des cases à cocher. Par ailleurs, il est possible de placer plusieurs formulaires au sein d'une même page, chacun possédant son propre bouton de validation.

Le paramètre `ACTION` indique au navigateur quel URL indiquer dans la requête qu'il devra formuler au serveur lorsque l'utilisateur validera le formulaire. Il s'agit bien sûr d'un programme, qui devra récupérer l'information envoyée par le navigateur et l'utiliser (voir 10.3 page 353).

```
<FORM ACTION="doc.cgi">

Indiquez la référence de la fenêtre qui vous intéresse
<BR>
<INPUT TYPE="text" NAME="remarque" SIZE=30>
<P>
Cocher cette case
<INPUT TYPE="checkbox" NAME="doc">
pour recevoir une documentation
<P>

<INPUT TYPE="submit" VALUE="cliquez ici pour valider le formulaire">

</FORM>
```

Figure 10.54 *Mise en place d'un formulaire HTML*

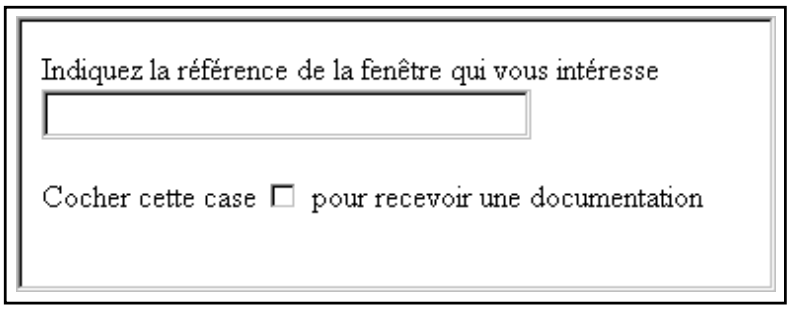

Figure 10.55 *Exemple de formulaire*

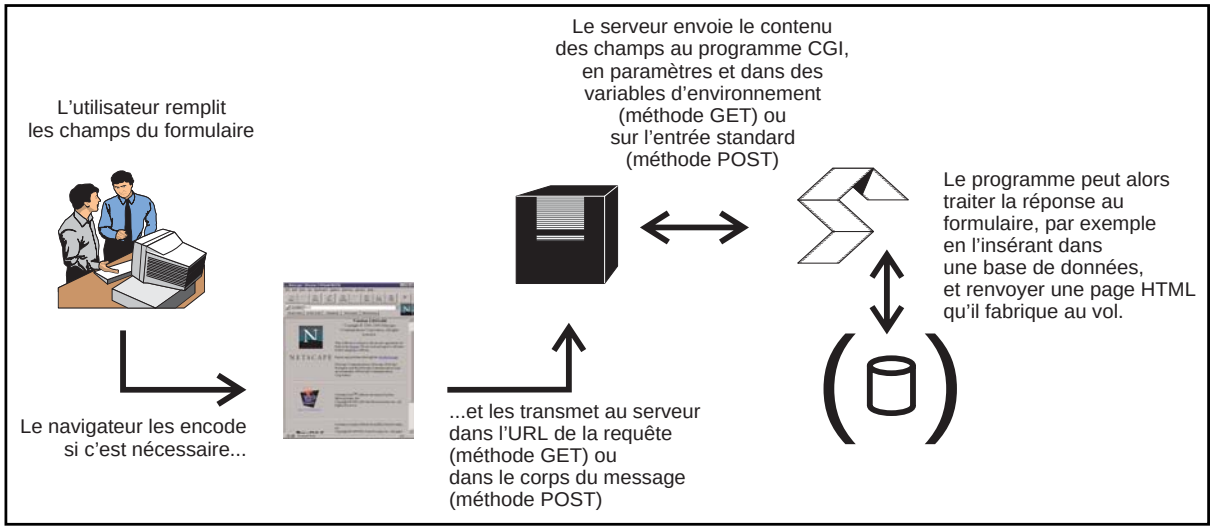

Figure 10.56 *Chaîne de validation d'un formulaire*

Liste des types de champs

Nous résumons ici quelques instructions utilisées pour créer des champs au sein d'un formulaire. Chacune accepte le paramètre `NAME` : c'est lui qui détermine le champ dans le résultat du formulaire. Les contenus des champs seront transmis au serveur sous la forme ***nom=valeur***.

Champ de texte

Utilisation :

```
<INPUT TYPE="text" NAME="nom du champ"
VALUE="valeur par défaut">
```

Cette instruction fait apparaître une case dans laquelle l'utilisateur peut saisir une ligne de texte. Pour faire apparaître une case permettant de saisir plusieurs lignes, il faudra utiliser l'instruction <TEXTAREA> ci-dessous.

Zone de texte

Utilisation :

```
<TEXTAREA NAME="nom du champ">
Texte par défaut
(éventuellement sur
plusieurs lignes)
</TEXTAREA>
```

La zone de texte fonctionne comme le champ de texte, mais permet de saisir plusieurs lignes. Les sauts de ligne seront encodés par le navigateur, ils ne posent donc pas de problème particulier.

Attention cependant : selon les systèmes et les navigateurs, les sauts de ligne peuvent se présenter sous la forme retour chariot + saut de ligne. Le programme CGI devra tenir compte de cette spécificité, et supprimer au besoin les caractères superflus.

Case à cocher

Utilisation :

```
<INPUT TYPE="checkbox" NAME="nom du champ"
VALUE="valeur renvoyée" CHECKED>
```

La case à cocher renvoie la valeur indiquée si elle a été cochée par l'utilisateur. Dans le cas contraire, elle n'apparaîtra pas dans les informations reçues par le serveur. Le paramètre `CHECKED` permet d'indiquer que la case doit être cochée par défaut.

Plusieurs cases à cocher peuvent porter le même nom ; toutes les valeurs correspondantes seront indiquées dans le résultat. On peut ainsi fabriquer des formulaires à choix multiple pour un même critère.

Bouton radio

Utilisation :

```
<INPUT TYPE="radio" NAME="nom du champ"
VALUE="valeur renvoyée" CHECKED>
```

Le bouton radio fonctionne exactement comme la case à cocher. Sa seule particularité est que si plusieurs boutons radio portent le même nom, seul un pourra être coché, tout comme avec le sélecteur de stations d'un autoradio.

Bouton de remise à zéro

Utilisation :

```
<INPUT TYPE="reset" NAME="nom du bouton"
VALUE="label du bouton">
```

Ce bouton permet de redonner à tous les champs leur valeur par défaut.

Bouton de validation

Utilisation :

```
<INPUT TYPE="submit" NAME="nom du bouton"
VALUE="label du bouton">
```

Ce bouton permet de valider le formulaire. Les données seront envoyées au serveur qui les transmettra le cas échéant au programme CGI spécifié dans le paramètre `ACTION` de l'instruction <FORM>.

10.3 Utilisation de programmes avec le Web

On pourrait qualifier le standard HTML de purement *documentaire* : il définit des règles pour la diffusion de documents et leur mise en page en fonction de leur structure, mais elle ne permet pas l'interativité qu'on peut obtenir avec les programmes informatiques habituels.

Cependant, il est possible d'interfacer les pages d'un serveur avec des programmes ou des applications chargées par exemple de conserver dans des bases de données les informations saisies par les utilisateurs sur des formulaires, ou encore de personaliser le contenu du serveur en fonction des goûts de ceux qui le consultent. On utilise pour cela la norme CGI (*Common Gateway Interface*), qui définit le format des échanges entre le serveur HTTP et les programmes externes auxquels il fait appel. Par abus de langage, on désigne souvent ces derniers par l'expression « scripts (ou programmes) CGI ».

Les spécifications complètes de la norme CGI ainsi que la rédaction de programmes permettant de lier un serveur à des bases de données ou à d'autres applications dépassent le cadre de cet ouvrage. Nous allons cependant en présenter les principaux concepts et évoquer les techniques utilisées par les serveur HTTP pour communiquer avec les programmes CGI.

10.3.1 La norme CGI

La technique utilisée pour permettre au serveur HTTP d'échanger des informations avec un programme externe a été formalisée à travers la norme CGI (*Common Gateway Interface*). Elle spécifie la nomenclature des paramètres échangés et leur codage. Son avantage principal est qu'elle est assez stricte et particulièrement répandue, ce qui permet aux concepteurs de sites Web de changer de logiciel serveur sans adapter leurs programmes. Ses défauts les plus évidents résident dans ses performances médiocres, principalement liées au fait que les programmes sont lancés à chaque requête, ce qui génère une charge machine importante et rend difficile le suivi d'une session utilisateur. Elle tend aujourd'hui à évoluer vers un second niveau, dénommé Fast-CGI, tout aussi portable mais plus efficace et mieux adapté au mode transactionnel, puisque les programmes CGI fonctionnent en permanence et sont sollicités par le serveur HTTP par l'intermédiaire de messages système ou TCP/IP. On trouve également sur le marché des outils intégrés, généralement construits autour d'un L4G et d'un SGBD, qui dispensent le concepteur d'utiliser des programmes externes, le code des applications étant placé directement au sein des pages. Bien entendu, ces outils sont le plus souvent très spécifiques.

Selon la norme CGI, le serveur HTTP communique avec les programmes qu'il lance par l'intermédiaire :

- de variables d'environnement ;
- de l'entrée standard ;
- des arguments de ligne de commande.

Ces informations sont par ailleurs transmises au serveur par le navigateur, au moyen :

- de l'URL ;
- des champs contenus dans l'en-tête des requêtes ;
- des corps de messages POST.

Quant au programme, tout ce qu'il 'imprime' sur la sortie standard est redirigé vers le client, généralement par l'intermédiaire du serveur qui se charge de compléter l'en-tête. Le programme CGI doit donc en particulier préciser, au moyen du champ `Content-type`, le type des informations qu'il renvoie. Pour cette raison, les premières lignes imprimées par un script CGI sont généralement les suivantes (le signe ¶ indique un saut de ligne).

```
Content-type: text/html¶
¶
```

Paramètres de l'URL

La forme globale d'un URL HTTP tel qu'interprété selon la norme CGI prend, en détaillant les différents éléments, la forme suivante.

```
http://machine/chemin/fichier/path_info?query_string
```

- `path_info` ne fait pas partie du chemin d'accès au fichier ou au script CGI, mais représente une information complémentaire qui sera transmise au programme CGI par l'intermédiaire de la variable d'environnement PATH_INFO. En recevant cet URL, le serveur va chercher le programme à exécuter en descendant dans les répertoires (à partir du répertoire Document Root, bien entendu). Lorsqu'il l'aura trouvé, il considérera que la fin du chemin d'accès est en réalité le `path_info`.
- `query_string` est une série d'arguments ou de couples `argument=valeur` séparés par le signe &, qui seront passés au programme CGI, comme arguments en ligne de commande pour les premiers, et comme variables d'environnement pour les seconds.

Ainsi, l'URL :

```
http://www.fenetre.fr/catalogue/liste.cgi/catalogue/lis
te.html?bdd\_1\&trier=oui\&chercher=Fen%EAtres+%E0+peti
ts+carreaux
```

peut se décomposer de la manière suivante.

Nom de machine	`www.fenetre.fr`
Chemin d'accès	`/catalogue/`
Programme CGI	`liste.cgi`
PATH_INFO	`/catalogue/liste.html`
QUERY_STRING	`bdd_1&trier=oui&chercher=Fen%EAtres+%E0+petits+carreaux`

On note que, conformément à la syntaxe générale des URL, la variable d'environnement QUERY_STRING a été encodée. Le signe & sert à délimiter les différents paramètres. Le

signe = sert à marquer un couple (paramètre, valeur). Le signe + indique un espace. La variable doit donc être interprétée comme suit :

```
bdd_1
trier=oui
chercher=Fenêtres à petits carreaux
```

La valeur simple `bdd_1` sera passée en argument. Les autres paramètres devront être récupérés par le programme CGI en décodant la variable d'environnement QUERY_STRING. La figure 10.57 page suivante montre un exemple de fonction Perl qui réalise ce travail et place les arguments dans le tableau `@params` et les autres paramètres dans le tableau associatif `%params`. Elle attend en entrée la valeur de la variable d'environnement QUERY_STRING.

On observe qu'il est possible d'obtenir plusieurs valeurs pour un même nom de paramètre. En effet, les champs du type *checkbox* peuvent être multiples, de même que les valeurs renvoyées par un champ du type *multiple select*. La fonction `request_decode` chaîne ces valeurs dans la même variable, en les séparant par des virgules.

Ce choix est bien sûr arbitraire, au besoin (si les valeurs elles-mêmes peuvent contenir des virgules) on pourra utiliser des caractères nuls (code ASCII 0), des retours chariot (code ASCII 13), etc. Dans cet exemple, l'utilisation de la virgule nous permet de mieux visualiser les paramètres.

On pourra tester ce programme en le plaçant dans un des répertoires de l'arborescence du serveur, par exemple `tests/decode.cgi`. Sur le serveur, nous pouvons demander URL de la forme suivante et observer le résultat (figure 10.58 page 358).

```
http://www.fenetre.fr/tests/bdd_1&trier=oui&trier=croiss
ant&chercher=Fen%EAtres+%E0+petits+carreaux
```

Champs de l'en-tête HTTP

Le serveur définit une série de variables d'environnement qui reflètent certains champs de l'en-tête de la requête formulée par le client. Bien entendu, la présence ou le contenu de ces variables dépend de la forme de la requête. Dans la pratique, il dépend aussi des fonctionnalités du client et du serveur.

Parmi les plus courantes, citons :

`AUTH_TYPE`

> Le type d'authentification effectuée par le client. Nous avons vu le type `Basic` section 9.1.6 page 286.

```
sub decode {
local ($string) = @_;
# Les + sont en fait des espaces
      $string =~ tr/+/ /;
# Décodage des groupes %xx où xx est le code ASCII du caractère
      $string =~ s/%([A-F0-9][A-F0-9])/pack("C", hex($1))/gie;
      $string;
}

sub request_decode {
local ($incoming)=@_;
local (@pairs,$name,$value);
# Initialisation des tableaux (variables globales)
  undef(@params),undef(%params);
# La chaîne de paramètres doit être découpée au niveau des signes &
  @pairs = split(/&/, $incoming);

# On examine chaque paramètre. Les arguments sont placés
# dans le tableau @params. Les autres paramètres sont placés
# dans le tableau associatif %params.
  foreach (@pairs) {

# S'il n'y a pas de signe = c'est un argument
    if (!/=/) {
      push(@params, &decode($_));

# Sinon, il s'agit d'un couple nom=valeur
    } else {
      ($name, $value) = split(/=/, $_);
      $name=&decode($name);
      $value=&decode($value);

# Une particularité : un paramètre peut avoir plusieurs valeurs.
# Dans ce cas, elles seront séparées par des virgules.
      if ($params{$name}) { $params{$name}.=",".$value; }
      else { $params{$name}=$value; }
    } } }

# Exemple d'utilisation
$query=$ENV{"QUERY_STRING"};
&request_decode($query);

# Affichage du contenu des variables.
# On envoie d'abord un Content-type dans l'en-tête du message.

print "Content-type: text/html\n\n";

print "<B>Arguments :</B><BR>";
print join("<BR>",@params);

# Les couples (nom,valeur) sont affichés dans un tableau :
# les noms dans la première colonne, les valeurs dans la seconde.

print "<P><B>Autres paramètres :</B><BR>";
print "<TABLE><TD>";
print join("<BR>",keys %params);
print "</TD><TD>";
print join("<BR>",values %params);
print "</TD></TABLE>";
```

Figure 10.57 *Programme Perl pour décoder les URL HTTP*

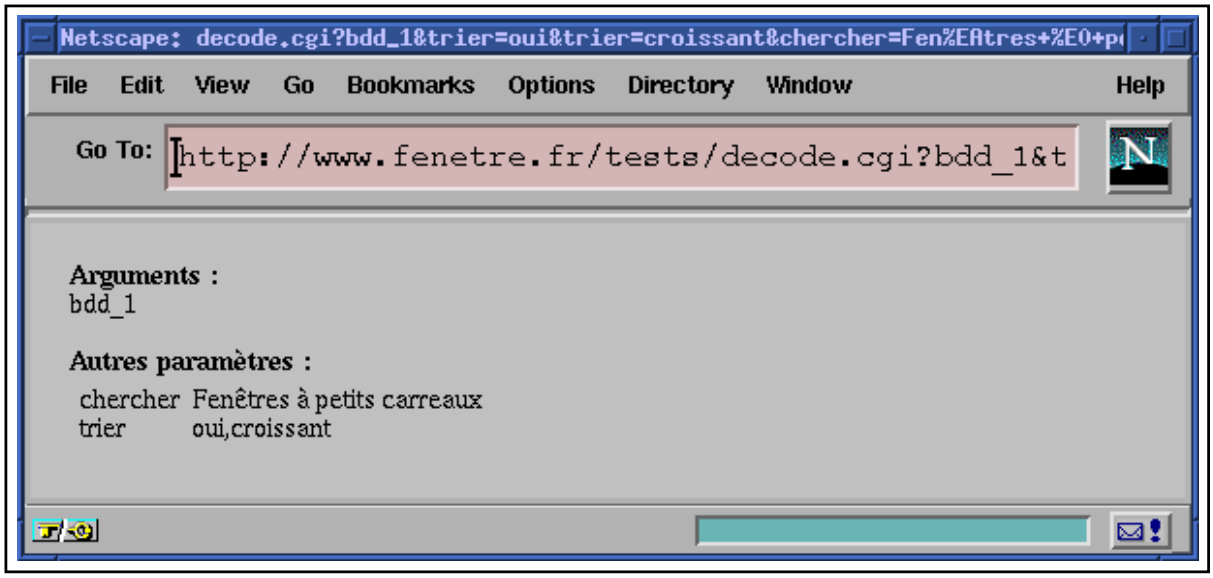

Figure 10.58 *Exemple d'utilisation du script de décodage d'un URL*

`REMOTE_USER`

L'identificateur de l'utilisateur, tel que fourni lors de l'authentification (le nom de *login*, en quelque sorte).

`REQUEST_METHOD`

La méthode utilisée pour la requête (le plus souvent GET ou POST).

Utilisation de la méthode POST

La variable d'environnement QUERY_STRING est principalement utilisée avec la méthode GET, puisque cette méthode précise que les différentes informations permettant de répondre à la requête doivent être transmises dans l'URL. Cette méthode est simple et facile à tester, puisqu'on peut fabriquer l'URL sans faire appel à un navigateur et une page HTML contenant un formulaire.

En revanche, lorsqu'il est nécessaire de tranférer des données dépassant un certain volume (quelques dizaines d'octets), il devient peu commode d'utiliser l'URL. La méthode POST répond à cette préoccupation, puisqu'elle permet de transférer les informations auparavant contenues dans la variable QUERY_STRING au sein du corps du message HTTP, et non plus dans l'en-tête. On peut ainsi expédier d'importants volumes de données (sans nécessairement les encoder puisqu'il suffit que le client indique le type du fichier dans le champ `Content-type`).

Dans la pratique, la méthode POST sera utilisée exactement comme la méthode GET, pour récupérer une réponse à un formulaire. Les informations seront simplement lues depuis l'en-

trée standard, comme le montre ce petit script *shell*, `post.cgi` :

```
#!/bin/sh

echo Content-type: text/plain
echo

cat -
```

On pourra le tester en l'appelant à partir de la page HTML suivante.

```
<FORM ACTION="post.cgi" METHOD=POST>
<INPUT TYPE="text" NAME="ligne de texte" size=40>
<P>
<TEXTAREA NAME="zone de texte" ROWS=6 COLS=40>
Texte par défaut
(sur plusieurs lignes)
</TEXTAREA>
<P>
<INPUT TYPE="submit" NAME="submit"
VALUE="Envoyer le formulaire">
</FORM>
```

Les champs du formulaire parviennent au programme CGI sous la même forme encodée qu'au sein des URL. Il est donc nécessaire d'utiliser la fonction `request_decode` pour décomposer les informations reçues sur l'entrée standard. En Perl, cela donnerait :

```
#!/usr/local/bin/perl

require "request_decode.pl";

$query=<>;
&request_decode($query);

[...]
```

Note : Il est tout à fait possible d'utiliser la méthode POST tout en insérant une information spécifique dans l'URL (par exemple au sein d'une instruction `<A HREF=...>`). L'entrée standard contiendra alors le corps du message transmis par le client, et la variable d'environnement QUERY_STRING jouera son rôle habituel, comme avec la méthode GET.

10.3.2 Application : affichage des variables d'environnement

Le serveur HTTP met à la disposition des programmes CGI qu'il lance un certain nombre de variables d'environnement. Ces variables contiennent, outre les différents paramètres que nous avons déjà évoqués, des informations plus spécifiques concernant, par exemple, le modèle et la version du serveur ou du client, le nom du script appelé, la valeur des chemins d'accès du serveur, et notamment le Document Root...

Pour afficher simplement ces informations, il suffit de rédiger puis d'exécuter le petit script *shell* suivant.

Il se contente d'exécuter la commande `set` pour afficher l'état des variables d'environnement. Avant tout, il indique bien sûr le type de données, ici du texte pur : `text/plain`.

```
#!/bin/sh

echo Content-type: text/plain
echo

set
```

Ce programme, que nous nommerons `environnement`, peut être placé dans le répertoire des scripts du serveur, par exemple :

```
/usr/local/httpd/cgi-bin
```

Il est appelé en demandant, par l'intermédiaire du navigateur – ou à l'aide de la commande `telnet` à destination du port du serveur – un URL de la forme :

```
http://www.fenetre.fr/cgi-bin/environnement
```

10.3.3 Application : cartes cliquables (clickable image maps)

La faiblesse principale des liens hypertexte lorsqu'on les applique à des images est que la forme de l'image elle-même est nécessairement rectangulaire. Il n'est donc a priori pas possible de concevoir des menus dont les éléments seraient de formes diverses, ou encore des plans ou cartes qui enverraient le lecteur vers des URL différents en fonction de l'endroit où il aurait cliqué.

Afin de répondre à ce type de besoin, le langage HTML définit la notion de carte cliquable (*clickable map*). Le principe en est simple, puisque lorsqu'il rencontre une image de ce type (signalée par le paramètre `ISMAP` dans l'instruction <IMG>) le navigateur repère les coordonnées de la souris par rapport au coin haut gauche de l'image lorsque l'utilisateur clique sur l'image en question. Ces coordonnées X,Y sont ajoutées en paramètre à la requête sous la forme décrite pour les appels CGI.

Le lien doit donc a priori pointer vers un script CGI capable d'interpréter les coordonnées en fonction de l'image, et de répondre par exemple par une redirection (champ `Location` dans l'en-tête HTTP) vers l'URL qui convient. Dans l'exemple ci-dessous, l'image est divisée en trois parties : le cercle et le rectangle pointent chacun vers un URL particulier, et le reste de l'image ne pointe vers rien (ou plus exactement, il pointe vers la page contenant l'image, afin que le navigateur n'affiche pas une erreur si on clique sur cette partie). Le programme

présenté figure 10.59 page suivante est rédigé en C, sa compréhension ne devrait poser aucun problème. La redirection est obtenue en fournissant simplement le champ `Location: [URL à charger]`; le reste de l'en-tête devrait être mis en forme par le serveur HTTP qui y ajoutera le statut approprié (code `302 : Moved Temporarily`).

Pour tester ce programme, il suffit de remplacer les URL indiqués au début du source par des URL valides et de compiler l'ensemble avec, par exemple :

```
gcc -o simplemap.cgi simplemap.c
```

Le programme doit être placé dans le répertoire réservé aux scripts (par exemple le répertoire correspondant à `/cgi-bin/`), ou porter l'extension `.cgi` et être installé dans le répertoire courant (ne pas oublier de configurer le serveur pour qu'il accepte d'exécuter les programmes placés dans les répertoires des documents).

La page HTML est très simple, puisqu'elle se résume à :

```
<A HREF=simplemap.cgi>
<IMG ISMAP SRC=rondrect.gif BORDER=0></A>
```

Bien entendu, il faut créer un fichier image nommé `rondrect.gif` et le placer dans le même répertoire que la page et le programme CGI. Lorsqu'on positionne la souris sur l'image, Netscape Navigator affiche dans la barre d'état de la fenêtre la position qui sera envoyée en paramètre au programme. Pour visualiser la transaction entre le navigateur et le serveur, on peut demander directement la page en se connectant par `telnet` sur le port du serveur.

Si le serveur n'est pas capable de détecter la redirection à partir du simple champ `Location` (ce qui est assez rare) cela signifie sans doute qu'il envoie la réponse du programme CGI telle quelle, sans analyser l'en-tête. Il faut alors envoyer le code 301 ou 302 soi-même.

Plutôt que d'indiquer une redirection pour la page elle-même (zone située en dehors du rond et du rectangle), on peut alors répondre par le code d'état (*status code*) `204 (No Content)` qui indique que le document est vide, et ainsi économiser un nouveau chargement de la page : la plupart des navigateurs ne feront rien, la page initiale restera affichée, ce qui est somme toute le comportement souhaité.

Pour simuler ce comportement avec un serveur type NCSA ou Apache, on peut donner au programme CGI un nom commençant par `nph-` (pour *non parsed header*). Le serveur se contentera alors de transmettre la réponse sans y toucher. Le programme devient alors celui de la figure 10.62 page 364.

La référence hypertexte se transforme bien sûr en <A HREF=nph-map.cgi>.

simplemap.c

```
#define URLDefaut "http://www.fenetre.fr/tests/testmap.html"

#define URLRond "http://www.fenetre.fr/tests/rond.html"

#define URLRect "http://www.fenetre.fr/tests/rectangle.html"

void defaut() {
    printf("Location: %s\r\n\r\n", URLDefaut);
    exit(0);
}

void URL1() {
    printf("Location: %s\r\n\r\n", URLRond);
    exit(0);
}

void URL2() {
    printf("Location: %s\r\n\r\n", URLRect);
    exit(0);
}

main(int ac, char **av) {
int     x, y, n;

/* On vérifie que les paramètres sont bien là.
 * Les coordonnées sont passées sous la forme x,y et devraient
 * se trouver dans le premier argument passé au programme.
 * En cas d'erreur, on renvoie l'URL de la page courante.
 */

    if (ac<2) defaut();

/* On tente d'extraire X et Y */

    n = sscanf(av[1], "%d,%d", &x, &y);
    if (n<2) defaut();

/* Le premier lien se trouve sur le cercle de centre (100,100)
 * et de rayon 20...
 */

    if ((x-100)*(x-100) + (y-100)*(y-100) <= 20*20) URL1();

/* Le second lien se trouve sur le rectangle
 * de coordonnées (200,10) (230,60)
 */

    if (x>=200 && x<=230 && y>=10 && y<=60) URL2();

/* Le reste de l'image renvoie l'URL par défaut */
    defaut();
}
```

Figure 10.59 *Exemple de programme C gérant une carte cliquable*

```
telnet www.fenetre.fr 80
Trying 192.168.22.34...
Connected to www.fenetre.fr.
Escape character is '^]'.
GET /tests/simplemap.cgi?220,55 HTTP/1.0

HTTP/1.0 302 Found
Date: Fri, 13 Sep 1996 22:03:10 GMT
Server: Apache/1.1.1
Location: http://www.fenetre.fr/tests/rectangle.html
Content-type: text/html

<HEAD><TITLE>Document moved</TITLE></HEAD>
<BODY><H1>Document moved</H1>
The document has moved <A HREF="http://www.fenetre.fr/tests/rectangle.html">here
</A>.<P>

</BODY>
```

Figure 10.60 *Transaction HTTP dans le cas d'une carte cliquable*

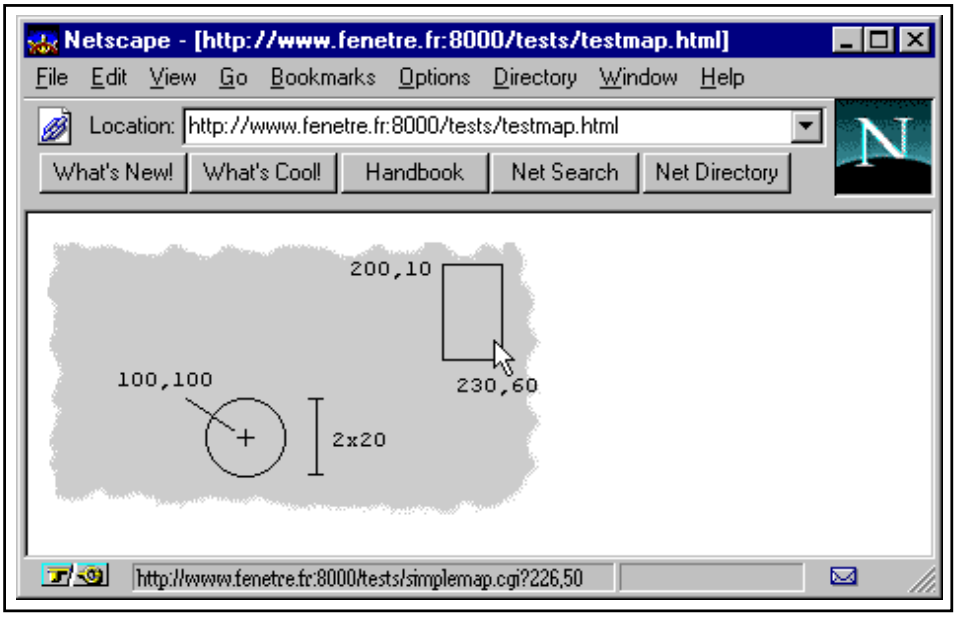

Figure 10.61 *Carte cliquable*

Utilisation de fichiers de paramètres

Le programme simplifié que nous avons vu au paragraphe précédent n'est pas d'un emploi particulièrement souple. On trouve sur le Net un programme en C beaucoup plus générique. C'est ce programme, `imagemap.c`, écrit par Rob McCool, qui est d'ailleurs à l'origine des fonctions de gestion des cartes cliquables intégrées aux serveurs récents : avec le serveur NCSA version 1.5 et suivantes, Apache, bien sûr, et la plupart des serveurs commerciaux tels ceux de Netscape et Microsoft il n'est plus nécessaire de faire appel à des programmes externes.

```
ls: $ cat nph-map.c

#define URLRond "http://www.fenetre.fr/tests/rond.html"

#define URLRect "http://www.fenetre.fr/tests/rectangle.html"

void defaut() {
    printf("HTTP/1.0 204\r\n\r\n");
    exit(0);
}

void URL1() {
    printf("HTTP/1.0 302\r\nLocation: %s\r\n\r\n", URLRond);
    exit(0);
}

void URL2() {
    printf("HTTP/1.0 302\r\nLocation: %s\r\n\r\n", URLRect);
    exit(0);
}

main(int ac, char **av) {
int     x, y, n;

/* On vérifie que les paramètres sont bien là.
 * Les coordonnées sont passées sous la forme x,y et devraient
 * se trouver dans le premier argument passé au programme.
 * En cas d'erreur, on renvoie l'URL de la page courante.
 */

    if (ac<2) defaut();

/* On tente d'extraire X et Y */

    n = sscanf(av[1], "%d,%d", &x, &y);
    if (n<2) defaut();

/* Le premier lien se trouve sur le cercle de centre (100,100)
 * et de rayon 20...
 */

    if ((x-100)*(x-100) + (y-100)*(y-100) <= 20*20) URL1();

/* Le second lien se trouve sur le rectangle
 * de coordonnées (200,10) (230,60)
 */

    if (x>=200 && x<=230 && y>=10 && y<=60) URL2();

/* Le reste de l'image revoie l'URL par défaut */

    defaut();
}

ls: $ gcc -o nph-map.cgi nph-map.c
```

Figure 10.62 *Version avec en-tête HTTP du programme simplemap.c*

Pour faciliter la mise en place des cartes cliquables, on enregistre les informations permettant de reconstituer les différentes zones de l'image dans un fichier de paramètres qui indique la forme et les coordonnées de chaque zone, et l'URL vers lequel elle renvoie.

Comme cela arrive souvent, les formats des fichiers de paramètres utilisés par les différents serveurs du marché se sont multipliés. Le plus courant est tout de même celui utilisé par le serveur NCSA. Il permet de définir des zones de forme rectangulaire, ronde ou polygonale, ainsi que des points de proximité. Chaque zone correspond à une ligne du fichier, qui précise dans l'ordre et séparés par des espaces : la forme, l'URL et les coordonnées (coins haut-gauche et bas-droite pour les rectangles, centre et un point de la circonférence pour les cercles, liste des sommets pour les polygones, ou encore coordonnées d'un point de proximité). On peut également indiquer un URL par défaut au cas où l'utilisateur cliquerait sur un point qui n'appartiendrait à aucune zone (ce qui est impossible si la forme point est utilisée, puisqu'il existera toujours un point le plus proche). Les éléments suivants correspondent à ceux illustrés figure 10.63.

```
rect http://www.fenetre.fr/solution1.html 10,10,40,40
circle http://www.fenetre.fr/solution2.html 100,30,120,30
poly http://www.fenetre.fr/solution3.html 50,50,34,86,61,79,109,90
default http://www.fenetre.fr/solutions.html
```

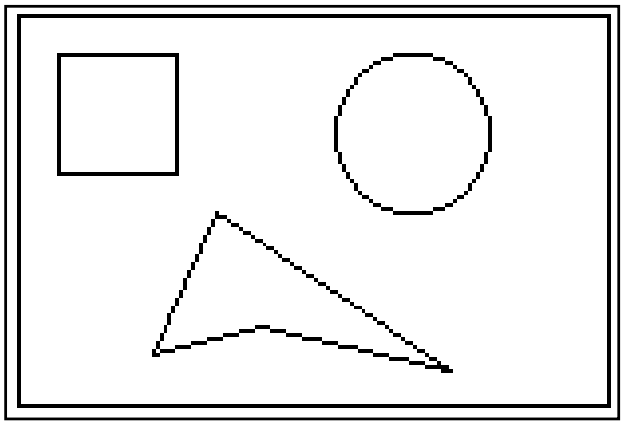

Figure 10.63 *Zones définies par les éléments d'une carte cliquable*

On utilisera par exemple les points de proximité pour des cartes géographiques. Ainsi pour la carte des villes de France, l'URL retenu sera celui du point le plus proche de l'endroit où l'utilisateur a cliqué.

```
point http://www.fenetre.fr/agences/paris.html 100,100
point http://www.fenetre.fr/agences/lyon.html 140,220
```

Ces fichiers de paramètres portent généralement l'extension `.map`.

Ils sont lus et interprétés par le serveur qui procède lui-même à une redirection pour indiquer au navigateur quelle page télécharger.

Configuration d'Apache

Pour qu'Apache prenne en compte les fichiers de paramètres des cartes cliquables, il faut indiquer dans la configuration que les fichiers portant par exemple l'extension `.map` doivent être traités par le module `mod_imap` (dont le rôle est précisément de lire le fichier de paramètres et d'interpréter la position envoyée par le navigateur en fonction des zones qui y sont décrites). Pour cela, on utilise l'instruction `AddHandler` en lui précisant en paramètre l'action `imap-file` correspondant au module interne de gestion des cartes cliquables, ainsi que l'extension des fichiers concernés (bien entendu, on pourrait choisir autre chose que `map`) :

```
AddHandler imap-file map
```

Le module `imap` d'Apache gère les fichiers au format NCSA décrit plus haut, avec une limite de 100 sommets par polygone et une souplesse supplémentaire puisqu'il sait traiter les URL relatifs. On peut donc lui fournir la ligne suivante :

```
point agences/paris.html 100,100
```

Il ajoutera au nom de la page la base de l'URL du fichier de paramètre. Si ce dernier se trouve en `http://www.fenetre.fr/villes.map`, le serveur renverra donc le champ `Location: http://www.fenetre.fr/agences/paris.html`.

Gestion côté client (client-side maps)

Les cartes cliquables ont un défaut majeur : il est parfois difficile pour l'utilisateur d'être sûr de l'URL demandé, puisque ce n'est pas cet URL qui s'affiche dans la barre d'état du navigateur, comme ce serait le cas avec un lien hypertexte classique, mais l'URL du fichier de paramètres ou éventuellement du programme CGI.

Afin de remédier à cette faiblesse, Netscape a introduit la notion de carte cliquable gérée par le navigateur lui-même, qu'on pourrait qualifier de *carte-client* (*client-side map*). Le fonctionnement est similaire à celui obtenu avec les cartes cliquables habituelles, mais le navigateur peut afficher le véritable URL de destination en fonction de la position de la souris.

Les paramètres de la carte doivent être traduits en HTML, grâce à une série de nouvelles instructions. <MAP> déclare la carte en précisant son nom, à la manière d'une ancre. <AREA> définit une zone, comme le font les lignes des fichiers de paramètres côté serveur. Un paramètre supplémentaire, USEMAP, permet d'indiquer dans l'instruction <IMG> qu'il existe une carte-client et de préciser l'endroit où elle se trouve. On notera que, s'il est possible d'insérer la carte dans la page qui contient l'image, ce n'est pas indispensable. Les cartes sont désignées comme les ancres, sous la forme `URL#NOM`. Il est ainsi possible de toutes les re-

grouper dans un répertoire particulier de l'arborescence, voire même dans un seul fichier, ce qui peut faciliter leur gestion :

```
<IMG SRC="villes.gif"
     USEMAP="/cartes/client.html#rondrect">
```

La traduction d'un fichier de paramètres en carte-client est assez simple, il suffit de changer la répartition des champs et d'insérer les instructions HTML. Cependant, toutes les formes ne sont pas encore disponibles. À titre indicatif, les versions 2.x de Netscape Navigator ne reconnaissent que la forme *rectangle*, il est donc conseillé de représenter les autres zones par un rectangle qui les contienne.

```
Version NCSA

circle http://www.fenetre.fr/tests/rond.html 100,100,120,100
rect http://www.fenetre.fr/tests/rectangle.html 200,10,230,60
default http://www.fenetre.fr/tests/testmap.html

version HTML

<MAP NAME=rondrect>
<AREA SHAPE="RECT" COORDS="80,80,120,120"
      HREF="http://www.fenetre.fr/tests/rond.html">
<AREA SHAPE="RECT" COORDS="200,10,230,60"
      HREF="http://www.fenetre.fr/tests/rectangle.html">
<AREA SHAPE="RECT" COORDS="0,0,249,139"
      HREF="http://www.fenetre.fr/tests/testmap.html">
</MAP>
```

Figure 10.64 *Traduction d'une carte NCSA en carte HTML*

Dans l'exemple présenté sur la figure 10.64, on a traduit la destination par défaut par une zone couvrant toute l'image. Les zones étant évaluées dans l'ordre dans lequel elles sont précisées, les deux premières zones seront tout de même prises en compte. Cependant, la destination par défaut est en fait la page elle-même : autant éviter de la recharger. Pour cela, il suffit de remplacer la destination (HREF="...") par le paramètre `NOHREF` qui indique qu'aucune action ne doit être effectuée :

```
<AREA SHAPE="RECT" COORDS="0,0,249,139" NOHREF>
```

En fait, Netscape définit l'action par défaut comme de ne rien faire. On peut donc tout simplement supprimer cette ligne.

Attention : une erreur commune consiste à remplacer purement et simplement la carte cliquable par une carte-client. Il ne faut pourtant pas oublier que de nombreux utilisateurs ne disposent pas d'un navigateur dernier cri. Comme toutes les autres instructions HTML, les instructions qui définissent la carte client sont ignorées par le navigateur s'il ne les comprend pas ; mais l'utilisateur risque de ne plus pouvoir accéder aux pages auxquelles elles faisaient référence. Il faut donc conserver les *deux* mécanismes, comme le montre la figure 10.65 page suivante.

```
<A HREF=simplemap.cgi>
<IMG ISMAP USEMAP=\#rondrect SRC=rondrect.gif BORDER=0></A>

<MAP NAME="rondrect">
<AREA SHAPE="RECT" COORDS="80,80,120,120"
      HREF="http://www.fenetre.fr/tests/rond.html">
<AREA SHAPE="RECT" COORDS="200,10,230,60"
      HREF="http://www.fenetre.fr/tests/rectangle.html">
</MAP>
```

Figure 10.65 *Syntaxe des cartes client*

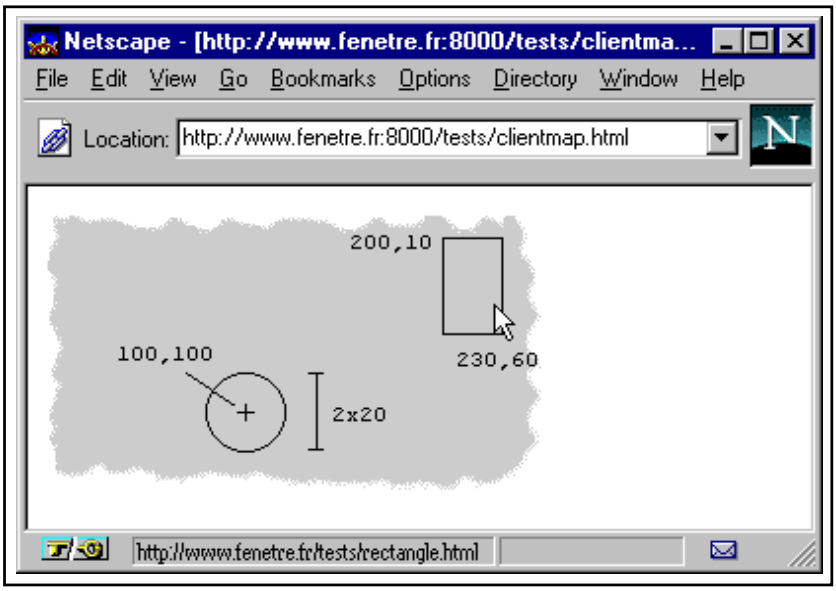

Figure 10.66 *Utilisation d'une carte cliquable*

10.4 Automatiser le formatage du contenu avec Apache

Le serveur Apache permet d'automatiser de manière intéressante une partie des tâches afférentes à la mise en place et la maintenance d'un serveur. On pourra ainsi insérer dans toutes les pages des en-têtes et pieds de page ou des menus contenus dans des documents HTML séparés, ou encore n'importe quelle autre information fournie par un programme CGI.

On pourra également appliquer un filtre aux documents d'un certain type en ajoutant un gestionnaire (*handler*) pour l'extension de ces documents, voire ajouter des modules directement intégrés au serveur. La plupart de ces opérations dépassent le cadre de cet ouvrage, mais nous allons tout de même évoquer quelques applications utiles des possibilités du serveur.

10.4.1 En-têtes et pieds de page

Très souvent, les en-têtes et pieds de page des documents du serveur varient peu d'une page à l'autre. On pourra en réaliser deux ou trois versions pour différentes parties du site, qu'on

placera dans des fichiers HTML séparés, **sans toutefois y ajouter les instructions de début/fin de document** (<HTML>, <HEAD> ou <BODY>). Un choix judicieux serait de les regrouper dans un répertoire dédié, par exemple `includes/` sous le répertoire racine du site (le *DocumentRoot*).

10.4.2 Lancement de scripts selon le type des fichiers

L'utilisation des fichiers inclus simplifie la tâche de maintenance, dans la mesure où il n'est plus nécessaire de systématiquement recopier le code HTML correspondant aux éléments fixes des pages. Cependant, elle ne dispense pas de mentionner l'instruction qui indique au serveur le fichier à inclure.

Une solution plus élégante serait d'inclure systématiquement des lignes de code dans toutes les pages. Ou mieux encore, d'inclure certaines lignes de code dans certains documents en fonction de leur extension de fichier.

Pour cela, on peut définir des gestionnaires particuliers pour les extensions choisies. Par exemple, nous retiendrons l'extension `.mhtml` pour repérer les documents auxquels on doit ajouter un menu en bas de page. Pour que les fichiers portant cette extension soient correctement traités, il faut y associer une action :

```
AddHandler ajouter_menu .mhtml
```

Il faut également définir l'action `ajouter_menu` afin que le serveur sache quelles mesures prendre lorsqu'on lui demande un fichier portant l'extension `.mhtml`. Ici en l'occurrence il s'agira d'exécuter un petit script *shell* chargé de compléter la page en lui adjoignant les quelques lignes correspondant au menu.

```
Action ajouter_menu /cgi-bin/menu.sh
```

Il est important de comprendre ce que fera le serveur lorsqu'on lui demandera un document portant l'extension `.mhtml`. Très simplement, il transformera cette requête en une autre requête portant sur le script précédemment défini, en y ajoutant le nom du document demandé, et c'est cette autre requête qu'il servira. Ainsi :

```
GET /tests/page.mhtml
```

deviendra

```
GET /cgi-bin/menu.sh/tests/page.mhtml
```

On reconnaît la forme d'URL spéciale déjà évoquée, fondée sur un *faux* chemin d'accès. Lorsqu'il évaluera l'URL, le serveur trouvera un exécutable `/cgi-bin/menu.sh` mais bien entendu pas de répertoire du même nom. Il arrêtera donc sa recherche à ce niveau, et le reste du chemin (ici `/tests/page.mhtml`) sera conservé dans la variable d'environnement PATH_INFO, que le script pourra utiliser, en n'oubliant pas de faire précéder sa valeur de celle de la variable DOCUMENT_ROOT, puisque PATH_INFO pointera vers un chemin d'accès relatif à la racine du serveur (et non du disque).

Afin de simplifier la tâche du script en question, on supposera que les instructions </BODY> et </HTML> ne figurent pas dans les pages et qu'elles sont automatiquement ajoutées.

menu.sh

```
\#!/bin/sh
echo Content-type: text/html
echo
cat $DOCUMENT\_ROOT/$PATH_INFO
cat <<FIN
 <P>
 <CENTER>
  <A TARGET=page HREF=http://www.fenetre.fr/cartes/menu.map>
 <IMG BORDER=0 ISMAP
      USEMAP=http://www.fenetre.fr/cartes/client.html\#menu
      SRC=menubar.gif WIDTH=562 HEIGHT=20></A>
 </CENTER>
 </BODY>
 </HTML>
FIN
```

Figure 10.67 *Ajout automatique d'un pied de page*

Pour appliquer le filtre à toutes les pages HTML, on utilisera le type MIME `text/html`. Dans ce cas il n'est plus nécessaire de définir un gestionnaire avec `AddHandler`, on peut directement associer l'action au type MIME :

```
Action text/html /cgi-bin/menu.sh
```

11

Relais et caches

Au fur et à mesure que l'Internet se développe, les voies de communications deviennent de plus en plus rapides mais le nombre d'utilisateurs augmente d'une manière telle que le modèle traditionnel de client-serveur ne suffit plus à assurer un temps de réponse acceptable dans bien des cas. Des analyses de trafic montrent que, dans de nombreux cas, les mêmes documents circulent de nombreuses fois et contribuent à la saturation des lignes. C'est pourquoi la communauté Internet est en train de mettre en œuvre de nouvelles techniques basées sur la réplication de l'information.

11.1 Principes de fonctionnement

11.1.1 Lecture directe

Lorsqu'un utilisateur de l'Internet veut accéder à une information dont il sait où elle se trouve, il utilise en général un programme appelé « client » qui se connecte à un autre programme appelé « serveur ». Le client fait sa demande au serveur, qui lui renvoie l'information voulue. La figure 11.1 montre comment un utilisateur accède à la page WWW se trouvant à l'URL `http://www.fenetre.fr/`.

Figure 11.1 *Accès direct à une page WWW*

Or certaines machines n'ont pas, pour des raisons de sécurité, un accès direct à l'Internet ; la plupart des entreprises disposant de données sensibles sur leur réseau informatique préfèrent n'exposer qu'un nombre limité de machines, tout en installant un service de relais pour accéder à l'information.

11.1.2 Passage par un serveur proxy

Un serveur **proxy** (ou relais) est un serveur qui, comme indiqué sur la figure 11.2, relaye les demandes faites par certains clients vers certains serveurs. Ici, la machine cliente, nommée `client.societe.fr` n'a pas d'accès direct à l'Internet ; c'est pourquoi elle utilise la machine relais `proxy.societe.fr` qui va faire à sa place la demande vers le serveur distant `www.fenetre.fr` et lui renvoyer la réponse.

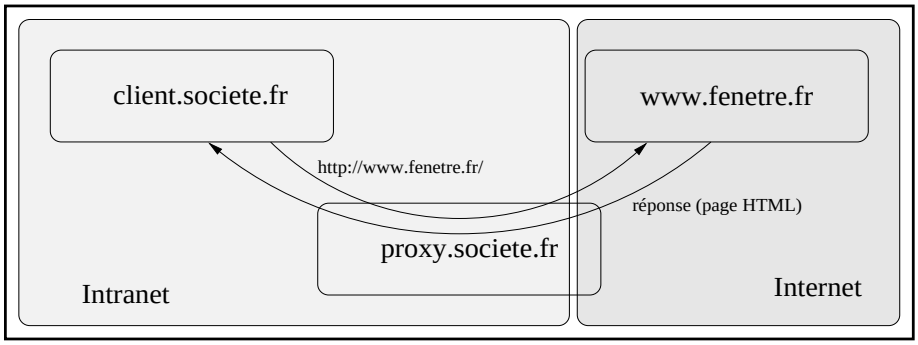

Figure 11.2 *Utilisation d'un relais*

Cette méthode résoud les problèmes d'accès à un serveur auquel on n'accèderait pas en temps normal, mais n'améliore en aucun cas les temps de transfert lorsque les liaisons entre le domaine `societe.fr` et le domaine `fenetre.fr` sont engorgés. C'est ici qu'intervient la notion de cache.

11.1.3 Mettre des documents dans le cache

Certaines données accessibles sur l'Internet présentent un caractère relativement statique ; les archives vieilles d'un an d'une liste de diffusion par exemple ne changeront *a priori* jamais, la liste des produits proposés par une société ne changera probablement pas tous les jours. Or, lorsque deux utilisateurs se trouvant dans le domaine `fenetre.fr` demandent à accéder à la même page (la page « Net Search » de `Netscape` par exemple) et recoivent par conséquent la même information, la page en question transite deux fois sur le réseau.

La figure 11.3 page suivante explique comment un serveur capable de conserver l'information dans un cache (« cache » est à comprendre ici comme « cache mémoire ») permet de réduire les temps de transfert dès le deuxième accès à la même page :

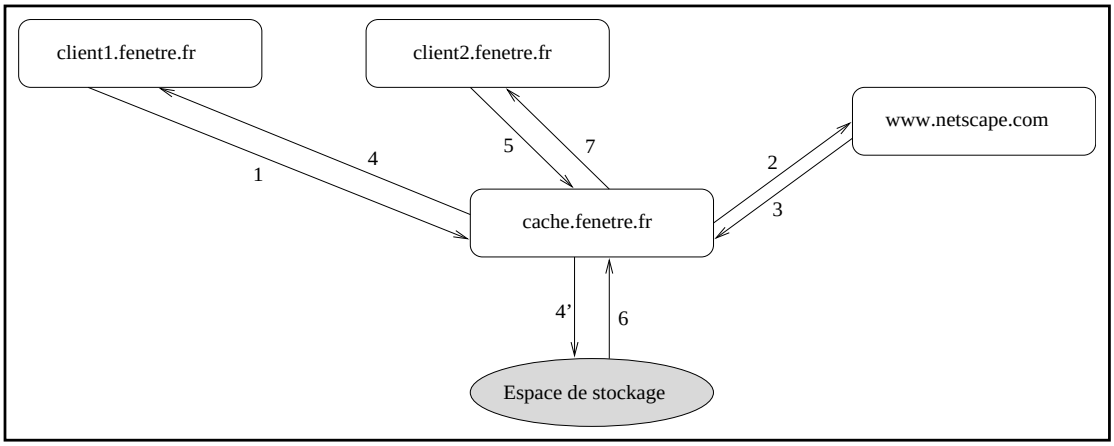

Figure 11.3 *Sauvegarde de l'information*

1. `client1.fenetre.fr` demande à accéder à la page principale de `Netscape`, c'est-à-dire `http://www.netscape.com/`. Sa requête est envoyé directement au serveur proxy-cache `cache.fenetre.fr` ;
2. le serveur envoie la requête à `www.netscape.com` ;
3. `www.netscape.com` renvoie la page demandée à `cache.fenetre.fr` ;
4. `cache.fenetre.fr` renvoie la page au demandeur (`client1.fenetre.fr`) et, en même temps, la sauve localement afin de pouvoir l'utiliser à nouveau en cas de nouvelle demande ;
5. `client2.fenetre.fr` souhaite également accéder à cette URL et envoie donc la demande à `cache.fenetre.fr` ;
6. le serveur proxy-cache s'aperçoit qu'il a déjà la page en local, et donc la recharge depuis son espace de sauvegarde ;
7. la page précédemment sauvée est renvoyée à `client2.fenetre.fr`, sans avoir transité sur le lien transatlantique.

Comme on le voit immédiatement, un serveur cache est obligatoirement un serveur proxy ; pour qu'il puisse détecter le fait que les mêmes pages sont demandées plusieurs fois, il faut qu'il ait connaissance de ces demandes. De plus, ce modèle ne s'applique pas qu'au WWW mais également aux autres protocoles basés sur le mécanisme de question/réponse tels que FTP ou Gopher.

11.1.4 Hiérarchie de serveurs intermédiaires

Cette technique de caches permet de réduire sensiblement le temps d'accès aux données distantes, mais elle peut encore être perfectionnée. Prenons le cas où le domaine `fenetre.fr` est en fait composé de deux sous-domaines dont les noms sont `ventes.fenetre.fr` et `materiel.fenetre.fr`, correspondant respectivement aux services « Ventes » et « Gestion du matériel » de la même entreprise, situés dans deux villes différentes et reliés par

une liaison RNIS. Il est avantageux pour chaque département d'avoir un serveur proxy-cache installé localement, afin de bénéficier des meilleurs temps d'accès possibles.

Toutefois, les temps d'accès entre les deux services sont relativement corrects, et il est donc dommage d'avoir deux systèmes complètement séparés, au risque de voir toute l'information transiter en double (une fois pour chaque sous-domaine). C'est pourquoi les caches des deux départements peuvent être configurés (voir figure 11.4) pour s'interroger l'un l'autre. On dit alors qu'ils sont **voisins**. De plus, ils s'adressent tous les deux à un serveur proxy-cache global pour l'ensemble de la société, qui devient donc leur **parent**.

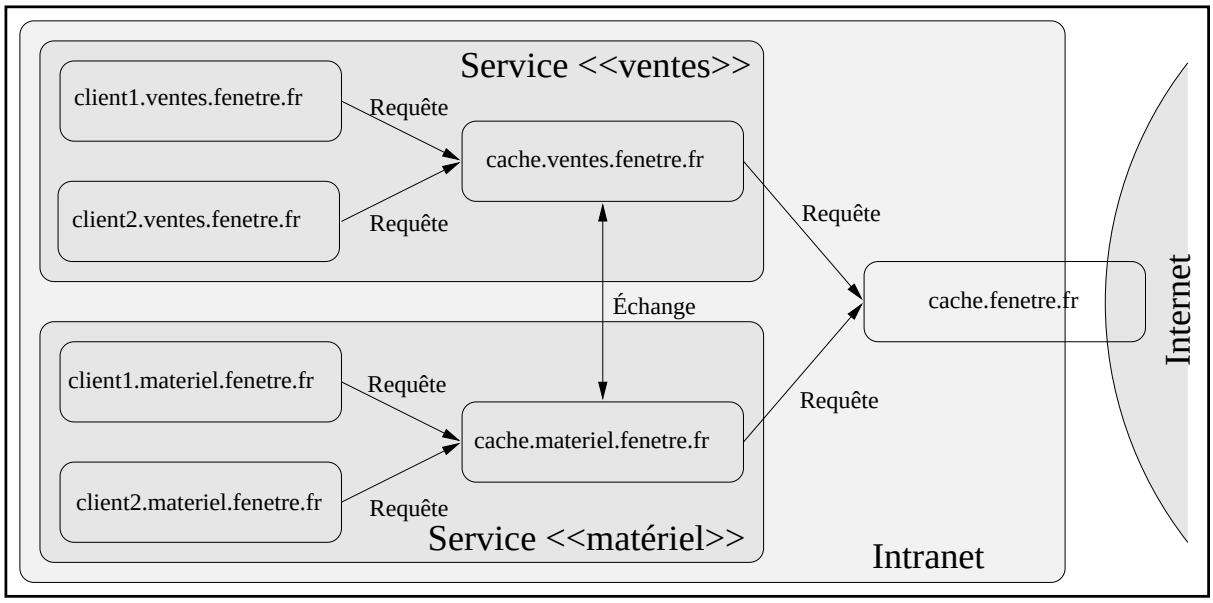

Figure 11.4 *Famille de caches*

11.2 Configuration d'un serveur proxy-cache

Le logiciel proxy-cache le plus avancé à l'heure actuelle s'appelle `squid`. Il est à la fois gratuit et performant, permet de relayer plusieurs protocoles et est prévu pour s'insérer dans une hiérarchie de caches.

11.2.1 Installation

Le programme `squid` peut être téléchargé sur plusieurs sites français, par exemple à l'URL `ftp://ftp.oleane.net/pub/mirrors/www/Squid/`. Étant donné qu'il se configure à l'aide d'`autoconf`, sa mise en place est un jeu d'enfant :

```
% gunzip -c squid-1.1-src.tar.gz | tar xpBf -
% cd squid-1.1
% ./configure

[...]

% make

[...]

% make install
```

11.2.2 Configuration

Le fichier de configuration de `squid` se trouve par défaut à l'endroit suivant (si aucune option contraire n'a été donnée à `configure`) : `/usr/local/etc/squid.conf`.

Il comporte de nombreux paramètres qu'il va falloir modifier pour les adapter à nos besoins. Dans le cas présent, nous allons configurer `squid` comme proxy-cache autonome, afin d'améliorer les performances d'accès depuis notre domaine `societe.fr`.

Directives globales

Les directives influant sur la configuration globale de `squid` sont présentées dans le tableau 11.1. Il est conseillé de conserver les valeurs par défaut, surtout lorsqu'on prévoit de s'insérer par la suite dans une hiérarchie de caches : il est toujours plus facile pour les voisins d'utiliser partout les mêmes paramètres que de les adapter pour chaque site.

Directive	Valeur conseillée	Explication.
`http_port`	`3128`	Port sur lequel se connecteront les clients.
`tcp_incoming_address`	`0.0.0.0`	Adresse d'interface sur laquelle écouter les requêtes.
`local_domain`	`fenetre.fr`	Domaine pour lesquels on n'utilise jamais de voisin.
`local_ip`		Idem.

Tableau 11.1 *Directives globales de configuration*

Gestion des ressources

Étant donné qu'il sauvegarde certaines données localement, `squid` est un gros consommateur de ressources système, notamment en termes de mémoire et de place disque.

Pour cela, il est possible de configurer de manière fine (voir tableau 11.2 page suivante) les ressources que `squid` est autorisé à utiliser.

Directive	Valeur conseillée	Explication.
`cache_mem`	16	Taille maximum de mémoire occupée par `squid` à un moment donné.
`cache_swap`	100	Idem, mais pour le disque.
`cache_swap_high`	90	Cette valeur correspond au pourcentage par rapport à `cache_swap` à partir duquel `squid` commencera à libérer de la place disque.
`cache_swap_low`	75	Pourcentage à partir duquel l'algorithme de nettoyage cité ci-dessus s'arrête.
`cache_mem_high`	90	Comme `cache_swap_high`, mais pour la mémoire.
`cache_mem_low`	75	Comme `cache_swap_low`, mais pour la mémoire.
`cache_dir`		Répertoire dans lequel seront sauvés les fichiers (cette directive peut être présente plusieurs fois).

Tableau 11.2 *Gestion des ressources*

Droits d'accès

Les droits d'accès sont gérés dans `squid` par le principe des ACL (Access Control List), permettant de décrire de manière précise les droits individuels. Chaque ACL permet de décrire une particularité de l'URL demandée, du client, du serveur, etc., et les ACL peuvent ensuite être combinées de manière logique afin de former des listes de droits d'accès et d'interdictions. Ces combinaisons seront utilisées dans des structures comme :

```
http_access allow combinaison1
http_access deny combinaison2
http_access allow all
```

Dans cet exemple, ce qui vérifie `combinaison1` est autorisé, ce qui vérifie `combinaison2` mais pas `combinaison1` est interdit, et tout le reste est autorisé.

Par exemple, si on combine une ACL autorisant l'accès à un serveur donné, l'ACL autorisant les ordinateurs des commerciaux à utiliser le proxy-cache et une ACL autorisant l'accès en dehors des heures de bureau, alors l'accès à ce serveur ne sera autorisé pour les commerciaux qu'en dehors des heures de bureau, à l'exclusion de qui que ce soit d'autre, de quelqu'autre serveur que ce soit et des heures de travail.

La syntaxe générale d'une ACL est :

```
acl nom type param1 ...
```

où `nom` est un nom attribué à cette ACL, `type` un type d'ACL et `param1` et la suite les paramètres supplémentaires. Ces paramètres dépendent du type choisi comme indiqué dans le tableau 11.3.

Type	Paramètres	Signification
`src`	`addresse/masque`	Adresse source (on peut également indiquer un intervalle en séparant deux motifs par un tiret).
`dst`	`addresse/masque`	Adresse de destination.
`srcdomain`	nom de domaine	Adresse source basée sur le nom du domaine.
`dstdomain`	nom de domaine	Adresse de destination basée sur le nom du domaine.
`time`	format date	Heures d'accès.
`url_regex`	`expression`	Expression régulière décrivant l'URL (voir les exemples fournis avec `squid`).
`urlpath_regex`	`expression`	Idem, mais sans le type de protocole.
`port`	`port`	Numéro de port.
`proto`	`protocole`	Nom de protocole (comme `HTTP`).
`method`	`méthode`	Nom de méthode (comme `POST`).

Tableau 11.3 *Types d'ACL*

Le format de date utilisé comme paramètre de l'ACL de type `time` est une lettre facultative désignant le jour de la semaine (voir tableau 11.4) suivie de deux heures séparées par des tirets. Par exemple, la chaîne « `MTWHF 9:00-17:00` » peut représenter les heures de travail de la semaine.

Symbole	Jour
`M`	Lundi
`T`	Mardi
`W`	Mercredi
`H`	Jeudi
`F`	Vendredi
`A`	Samedi
`S`	Dimanche

Tableau 11.4 *Jours de la semaine*

Voilà par exemple à quoi ressemble la partie « autorisation » du fichier de configuration de `cache.fenetre.fr` :

```
# Ces sites ne sont pas utiles au travail (ACL "detente")
acl detente url_regex ^http://www.detente.fr/ ^http://www.fun.edu/
# De manière générale, on n'autorise le proxy-cache que
# pour les machines de chez nous (ACL "nous")
acl nous srcdomain fenetre.fr
# Les heures de travail s'appellent "travail"
acl travail MTWHF 9:00-17:00
# Interdiction de tous les clients extérieurs
http_access deny !nous
# Autorisation de tous les sites (sauf ceux de detente) en permanence
http_access allow !detente
# Autorisation des sites de detente hors des heures de travail
http_access detente !travail
# Interdiction du reste
http_access deny all
```

11.2.3 Maintenance

La seule chose à faire après l'installation est la configuration du système pour que `squid` soit lancé automatiquement à chaque redémarrage de la machine. À ces fins, il existe le programme `RunCache`, livré avec `squid`, qui s'assurera, s'il est appelé au démarrage, qu'il y a toujours un processus `squid` prêt à répondre aux requêtes.

12

Multicast : le travail coopératif sur l'Internet

Depuis plusieurs années déjà, les utilisateurs de l'Internet tentent de travailler dans un environnement de plus en plus coopératif et ouvert. L'IETF (Internet Engineering Tasking Force) a, à l'origine, conduit deux expériences appelées « audiocast » dans lesquelles les chercheurs ont fait passer pour la première fois des données audio et vidéo en direct sur l'Internet. Depuis, la recherche dans ce domaine a énormément progressé, et a abouti à ce qu'on connaît maintenant sous le nom de « **multicast** ».

12.1 La communication point à point

L'Internet est un réseau basé sur l'utilisation de redondances au niveau du routage ; cela a été voulu par ses concepteurs pour pouvoir résister à la rupture de certaines de ses liaisons. Cependant, ces redondances n'avaient à l'origine qu'un seul rôle : celui de relier de manière fiable deux machines devant échanger des informations.

La figure 12.1 page suivante donne un exemple de communication point à point : un utilisateur doté d'un client WWW souhaite consulter le serveur de la société FeNETre Service Express afin de connaître les services qu'elle propose. Pour cela, il demande à son client WWW de se connecter sur la machine qui permet de consulter ces pages, à savoir `www.fenetre.fr`. Une connexion virtuelle, point à point, est établie entre ces deux ordinateurs, qui peuvent alors s'échanger l'information voulue.

De même, lorsque l'utilisateur enverra un message électronique au commercial de cette société afin de lui demander une livraison de matériel, les gestionnaires de courrier électronique

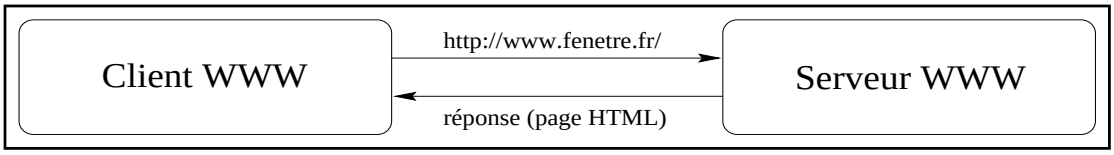

Figure 12.1 *Communication point à point*

établiront entre eux une liaison point à point afin de s'échanger, en utilisant par exemple le protocole SMTP, les informations nécessaires à la bonne transmission du message.

Cependant, ce mode de communication permet difficilement de simuler une salle de réunion dans laquelle travaillent plusieurs personnes ; même si les forums (voir le chapitre 6 page 219 à ce sujet) et les listes de diffusion permettent de s'adresser simultanément à un ensemble de personnes, on n'approche pas les conditions réelles des séances de *brainstorming*. Une possibilité de mise en place de cette notion de groupe interactif est présentée sur la figure 12.2 : un paquet, partant d'une machine (en noir), est diffusé vers plusieurs destinataires (représentés par des ronds grisés sur le schéma, les routeurs intermédiaires étant représentés par des ronds blancs).

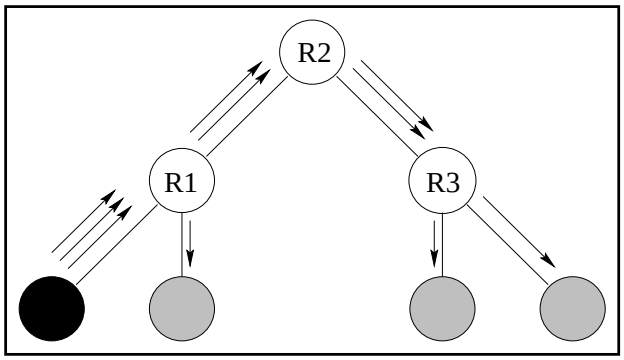

Figure 12.2 *Destinataires multiples*

Le principe est simple : chaque paquet IP est envoyé une fois à tous les destinataires. Malheureusement, il est totalement inefficace en termes de débit utilisé et de charge de travail pour les routeurs ; bien que les deux machines de droite se trouvent sur le même sous-réseau, les deux paquets qui leur parviennent suivent pratiquement le même chemin, ce qui signifie que l'information circule en double pratiquement de bout en bout. Il a donc fallu introduire pour l'Internet la notion de **groupe**.

12.2 Adresses de classe D et notion de groupe

Les machines connectées à l'Internet sont désignées par une adresse IP (ceci est expliqué page 33), qui appartient à l'une des trois classes d'adressage A, B ou C. Afin de pouvoir définir des groupes, on a donc introduit une nouvelle classe d'adresses, la classe D ou **adresses**

multicast.

Comme indiqué sur le schéma 2.4 page 36, une adresse de classe D commence toujours par la séquence de bits 1110. Cela signifie que les adresses IP commençant par un nombre compris entre 224 et 239 (inclus) désignent des adresses de groupes.

Si toutes les machines souhaitant dialoguer ensemble s'inscrivent dans un même groupe, le schéma de la figure 12.2 page ci-contre est remplacé par celui décrit figure 12.3. On voit aisément que toute la redondance existant précédemment a été supprimée ; l'information ne circule plus jamais en double, et n'utilise pas plus de bande passante que nécessaire. De plus, le routeur R4, qui n'a aucun besoin de recevoir l'information puisqu'aucune des machines de son réseau ne s'est abonnée au groupe, n'a rien reçu.

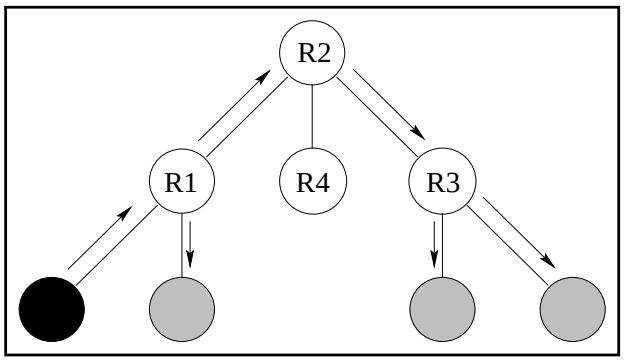

Figure 12.3 *Destinataires multiples appartenant à un même groupe*

Voici donc le fondement de l'IP multicast : une machine peut s'abonner à un ou plusieurs groupes multicast, et, à partir de cet instant, recevoir les paquets destinés à ce groupe. Notons de plus qu'il n'est pas nécessaire d'appartenir à un groupe pour pouvoir y envoyer des paquets.

12.2.1 Routage sur Ethernet

L'adressage de groupe existe sur Ethernet depuis sa création. Le principe est simple : comme expliqué sur la figure 12.4 page suivante, une adresse Ethernet spéciale, à savoir l'adresse représentée par `01-00-5E-00-00-00`, est combinée avec l'adresse de groupe multicast (25 bits de la première et 23 bits de la seconde) de manière à former une adresse de groupe Ethernet.

Lorsqu'une machine du réseau local demande à s'abonner à un groupe multicast, le noyau demande à ses cartes Ethernet de considérer les paquets contenant une telle adresse de destination comme « intéressants », et de les passer au noyau qui les exploitera. On peut noter que la totalité de l'adresse multicast n'apparaît pas dans l'adresse de groupe Ethernet, ce qui ajoute parfois certains déchets parmi les paquets reçus, déchets qui doivent ensuite être supprimés par le noyau.

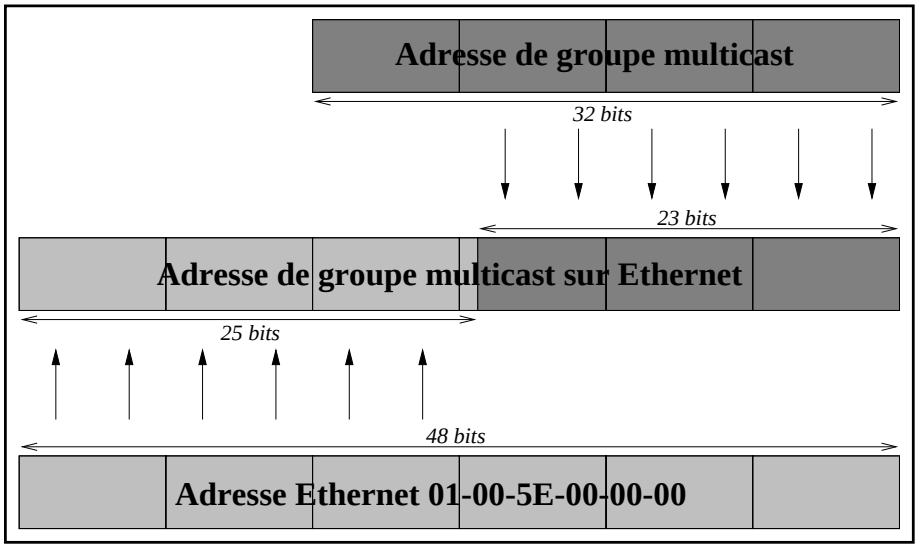

Figure 12.4 *Construction d'une adresse multicast sur Ethernet*

Le fait qu'Ethernet autorise l'adressage de groupe permet aux entreprises qui souhaitent utiliser le multicast pour le travail coopératif en interne de ne pas avoir à changer leur infrastructure ; au pire, il leur faudra obtenir de leurs fournisseurs une version récente des différents systèmes d'exploitation qu'ils utilisent, afin d'être sûrs que ceux-ci permettent d'utiliser IP multicast. Un noyau SunOS par exemple nécessite un certain nombre de modifications pour pouvoir utiliser le multicast, alors qu'un noyau Solaris ou Linux dispose des fonctions adéquates en standard.

12.2.2 Routage sur l'Internet : les routeurs multicast

Comme on l'a vu page précédente, un réseau local comme Ethernet permet l'adressage de groupe, résolvant ainsi le problème de la transmission des paquets multicast sur un brin du réseau. La transmission de tels paquets entre différents réseaux pose un problème plus complexe et nécessite des opérations de routage.

Il existe à ces fins plusieurs protocoles, que l'on peut classer en deux catégories :

1. **Les protocoles à mode dense :** ces protocoles supposent que la plupart des interfaces réseaux qu'ils gèrent voudront recevoir les paquets multicast. Les interfaces ne voulant pas en recevoir doivent (idéalement) constituer l'exception à la règle. Parmi ces protocoles, on peut citer DVMRP, MOSPF et PIM-DM.
2. **Les protocoles à mode dispersé :** ces protocoles, inversement aux précédents, considèrent que seules quelques interfaces (voire aucune) voudront, à un moment donné, recevoir les paquets multicast. Ce sont par exemple CBT ou PIM-SM.

Le choix du protocole dépend donc de l'utilisation typique du routeur ; si c'est un routeur d'entreprise, qui est donc en général connecté à un certain nombre de brins qui n'utiliseront

pas l'IP multicast, alors un protocole à mode dispersé sera le plus adéquat. À l'opposé, un routeur important reliant plusieurs fournisseurs d'accès à l'Internet aura intérêt, traditionnellement, à adopter un protocole à mode dense.

12.2.3 Routage sur l'Internet : les tunnels

Malheureusement, les routeurs présents sur l'Internet ne possèdent pas tous la possibilité de router des paquets IP multicast. Pour pallier ce manque et éviter que des zones séparées par un tel routeur ne puissent échanger de paquets multicast, on a adopté la notion de **tunnel** qui relie ces zones autonomes.

Un tunnel permet de transporter certains types de paquets à l'intérieur d'autres types de paquets. Dans le cas présent, pour compenser le fait que certains routeurs sont incapables de router des paquets multicast entre deux réseaux, on choisit une machine sur chacun d'eux pour lui confier le rôle de routeur multicast.

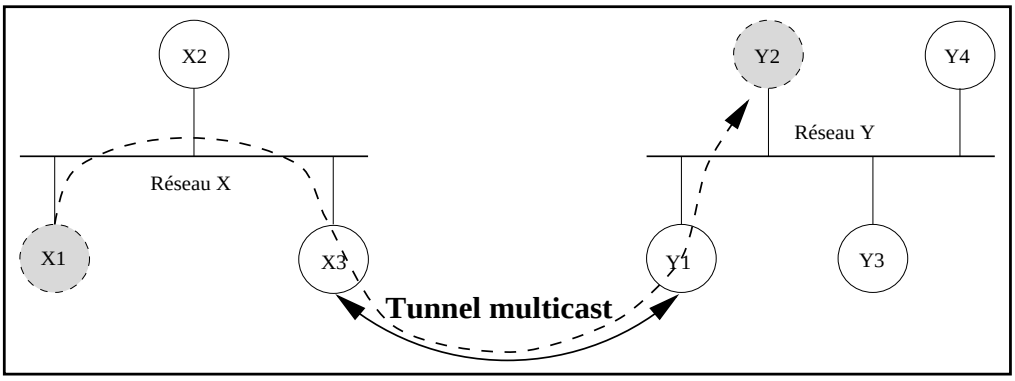

Figure 12.5 *Routage multicast utilisant un tunnel*

La figure 12.5 propose l'exemple de deux réseaux, appelés X et Y, sur lesquels circulent des paquets multicast. Afin de pouvoir faire transiter ces paquets entre les deux réseaux, on configure un tunnel entre les machines X3 et Y1 ; à chaque fois qu'un paquet multicast pour un groupe donné arrive sur le réseau X et qu'au moins une machine de Y est abonnée à ce groupe, la machine X3 envoie ce paquet à travers le tunnel après l'avoir encapsulé dans un paquet IP traditionnel (ce paquet utilise donc le routage IP habituel)[1]. De son côté, dès réception du paquet IP en provenance du tunnel, la machine Y1 le désencapsule et le réinjecte sur son réseau local Y. Ainsi, un paquet multicast émis par la machine X1 transitera par le réseau local X, la machine X3, le tunnel, la machine Y1, le réseau local Y et parviendra à la machine Y2 abonnée au groupe en question.

1. Dans les versions de multicast antérieures à mars 1993, on utilisait une autre possibilité offerte par IP, qui consistait à demander à ce que les paquets soient acheminés par un routage en mode « lâche » (*loose routing*). Ce protocole est encore en vigueur actuellement pour raison de compatibilité avec les anciens systèmes qui pourraient être encore en service, mais est voué à disparaître à terme à cause de ses piètres performances.

Grâce aux routeurs comprenant le protocole IP multicast et aux tunnels, on arrive à former sur l'Internet une structure multicast connexe et cohérente appelée le **MBone** (Multicast Backbone).

12.2.4 Deux protocoles de routage : DVMRP et MOSPF

Ces deux protocoles permettent de router des paquets multicast à travers le MBone, mais procèdent de deux manières différentes que nous allons examiner.

DVMRP

DVMRP, qui signifie *Distance Vector Multicast Routing Protocol*, est le protocole utilisé par le programme `mrouted` (voir page 387), bien que cette version apporte plusieurs améliorations notables par rapport au RFC 1075 qui le définit. Il utilise, comme le protocole RIP (voir page 51) la notion de vecteur de distance pour représenter sa connaissance du réseau MBone.

Malheureusement, comme RIP, il ne s'adapte pas facilement aux réseaux à grande échelle car la représentation par vecteurs de distance s'avère insuffisante pour représenter la topologie de manière précise et adéquate dans tous les cas de figure. Au-dessus de cette notion de vecteur de distance, DVMRP utilise un algorithme appelé *Truncated Reverse Path Broadcasting* qui lui permet de déterminer rapidement vers quelles interfaces (physiques ou virtuelles, comme les tunnels) il doit rediriger un paquet multicast entrant.

MOSPF

MOSPF est une extension multicast du protocole OSPF (voir page 52). Une version de MOSPF pour routeurs Proteon a été réalisée par John MOY, permettant à un ensemble de ces routeurs de s'échanger directement des paquets multicast sans passer par des tunnels, en économisant donc le coût de l'encapsulation et sans faire de copies de paquets inutiles.

Dans un futur proche, DVMRP sera intégré dans cette version de MOSPF, afin de permettre une collaboration complète entre ces deux formes de topologie et de réduire le trafic des messages multicast à l'intérieur d'un domaine.

12.2.5 Contrôle des paquets : durée de vie et seuil

Un problème essentiel se pose dans la définition du réseau MBone : comment éviter les boucles qui se traduiraient rapidement par une saturation des liaisons ? En fait, un modèle analogue à celui utilisé pour les paquets IP traditionnels a été adopté, la durée de vie ou **TTL** (Time To Live), auquel on a ajouté la notion de **seuil** (ou « *threshold* » en anglais).

Le TTL

Chaque paquet envoyé en multicast ou en IP traditionnel porte en lui une valeur numérique appelée TTL. Cette valeur représente pour un paquet, en nombre de « sauts » de routeur, le temps qui lui reste à vivre.

Ce nombre est décrémenté à chaque passage par un routeur ou tunnel. Lorsque la durée de vie arrive à 0, le paquet est supprimé car on considère qu'il est perdu.

Cela permet d'éviter qu'un paquet « immortel » ne circule indéfiniment sur le réseau en saturant les liaisons.

Le seuil

Le seuil est un nombre caractéristique pour un tunnel ou une interface réseau ; il représente la valeur minimum du TTL qu'un paquet doit avoir afin de pouvoir passer sur cette branche.

La figure 12.6 représente les différentes valeurs de seuil et leurs correspondances physiques.

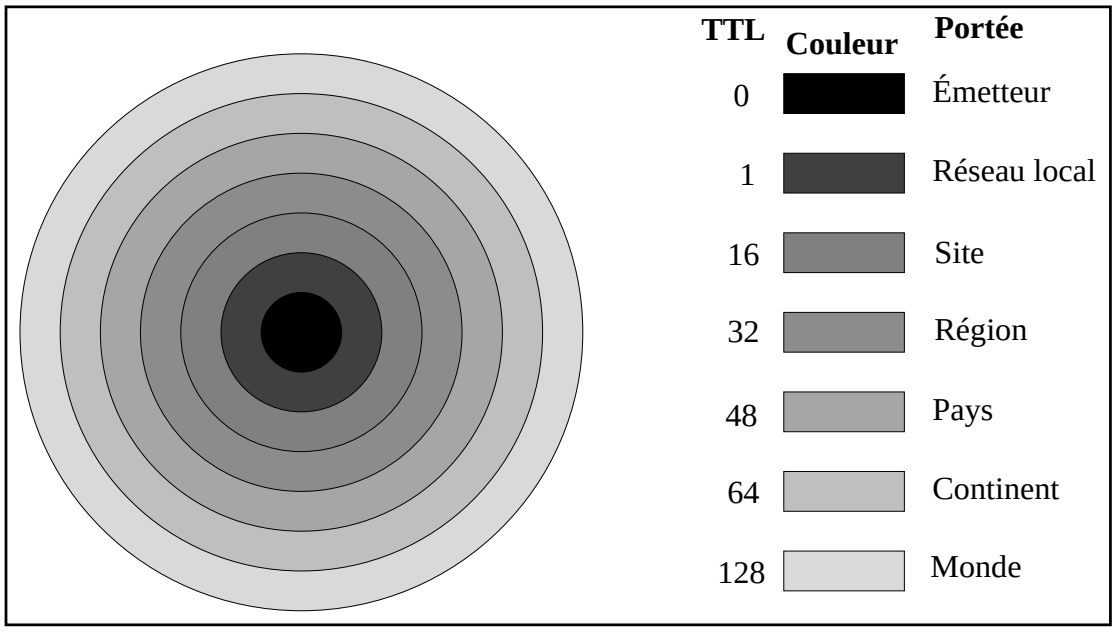

Figure 12.6 *Relation entre le TTL et la portée des paquets*

Comme indiqué sur cette figure, si, au moment d'arriver au routeur de notre entreprise, la valeur du TTL d'un paquet n'est pas d'au moins 16, alors le paquet ne sera pas envoyé à notre partenaire situé à l'autre bout du tunnel.

Cela signifie que si on configure les outils multicast pour qu'ils travaillent uniquement avec un TTL de 16, aucun paquet multicast ne sortira du site.

12.3 Connexion au MBone

12.3.1 Formalités administratives

Note importante

La première chose à faire si vous souhaitez connecter votre domaine au réseau MBone est de contacter votre fournisseur d'accès à l'Internet. En effet, s'il est lui-même connecté au MBone, c'est probablement à travers sa connexion que vous pourrez accéder au réseau multicast.

La partie française du réseau MBone est gérée depuis 1993 par Christian DONOT, qui en est le coordinateur. Pour demander à être intégré à la branche française du MBone, il suffit d'envoyer un courrier électronique à la liste de diffusion appropriée, dont l'adresse est `mbone-fr@inria.fr`[2]. Une réponse suivra, demandant des renseignements complémentaires sur la configuration de l'entreprise et donnant le nom et l'adresse IP d'une machine sur laquelle il sera possible de connecter un tunnel multicast.

Il est primordial d'arriver à un consensus en ce qui concerne la localisation d'un tunnel avant de le mettre en place ; en effet, le multicast évitant au maximum la propagation de paquets redondants, l'architecture du MBone doit suivre autant que faire se peut l'architecture physique du réseau.

12.3.2 Configuration d'un routeur Cisco

Les versions récentes des logiciels pour routeurs de marque Cisco permettent de faire du routage de paquets multicast, sous forme de tunnels (en effet, tant que l'ensemble des routeurs de l'Internet ne sont pas capables de router les paquets multicast, les tunnels restent indispensables pour maintenir la connexité du réseau MBone).

```
!
! Activation du routage des paquets multicast
!
ip multicast-routing
!
! Tunnel avec la machine fmroute1-1.exp.edf.fr
! (192.70.92.133)
!
interface Tunnel0
! Les paquets avec un TTL de moins de 16 restent en interne
ip multicast ttl-threshold 16
! Source: notre interface FAI, Destination: EDF
tunnel source 192.67.45.23
tunnel destination 192.70.92.133
! Le tunnel fonctionne en DVMRP
tunnel mode dvmrp
```

2. On peut s'inscrire sur cette liste en envoyant dans le corps du message la commande `subscribe` à l'adresse `mbone-fr-request@inria.fr`.

Les lignes ci-dessus présentent un exemple de tunnel configuré entre le routeur d'une entreprise et l'ordinateur qui lui a été indiqué par les responsables du réseau MBone français (ici la machine dont le nom complet est `fmroute1-1.exp.edf.fr`, à l'adresse `192.70.92.133`).

Dans cet exemple, `192.67.45.23` représente l'adresse de l'interface de notre routeur connectée à notre fournisseur d'accès internet.

Maintenant que notre routeur reçoit les paquets multicast, il faut établir un tunnel entre son interface interne (197.12.53.4) et une machine de notre réseau (197.12.53.18) qui recevra et retransmettra, à l'aide du programme `mrouted` (voir ci-dessous) les paquets multicast sur notre réseau local :

```
!
! Tunnel avec notre machine utilisant mrouted
! (197.12.53.18)
!
! Source: notre interface interne, Destination: notre machine
tunnel source 197.12.53.4
tunnel destination 197.12.53.18
! Le tunnel fonctionne en DVMRP
tunnel mode dvmrp
```

12.3.3 Configuration du programme mrouted

Le logiciel `mrouted` est disponible gratuitement[3] (binaires et sources). La distribution comprend plusieurs programmes :

map-mbone : permet d'afficher l'arbre des routeurs multicast accessibles depuis l'endroit courant ou depuis un endroit arbitraire ;

mrinfo : permet d'analyser la connectivité, et est très utile pour vérifier que les paramètres de routage multicast sont configurés correctement ;

mrouted : c'est le routeur, qui fera transiter les paquets du tunnel vers le réseau local et vice versa, ou vers un autre tunnel ;

mtrace : permet de faire l'équivalent du programme `traceroute`, mais sur un chemin multicast.

Ces programmes doivent être installés à la main, ainsi que leurs pages de manuel. Par exemple, sur Solaris, on peut faire :

```
% /usr/ucb/install -m 755 -o root -g other map-mbone mrinfo mrouted
    mtrace /usr/local/bin/
% /usr/ucb/install -m 644 -o root -g other map-mbone.8 mrinfo.8
    mrouted.8 mtrace.8 /usr/local/man/man8/
```

3. Voir l'URL `ftp://parcftp.xerox.com/pub/net-research/ipmulti/`.

Il reste à écrire le fichier de configuration de `mrouted`. Par défaut, `mrouted` cherchera ce fichier sous le nom `/etc/mrouted.conf`.

La configuration est assez proche de celle d'un tunnel pour un routeur. La contrepartie de l'exemple présenté page précédente est décrit ci-dessous :

```
#
# Nous souhaitons "arroser" notre réseau local sur
# l'interface physique le0.
#
phyint le0
#
# Tunnel vers notre routeur avec un seuil de 8
# (l'adresse interne du routeur est 197.12.53.4
# et celle de cette machine 197.12.53.18).
#
tunnel 197.12.53.18 197.12.53.4 metric 1 threshold 8
```

12.4 Les applications multicast

Un grand nombre d'applications multicast sont disponibles gratuitement sur l'Internet. Nous allons en détailler quelques-unes.

Note

Toutes les applications que nous décrivons ici (ainsi que beaucoup d'autres tout aussi utiles) sont disponibles librement sur l'Internet, à l'URL suivante : `http://www.merit.edu/net-research/mbone/.archive.html`.

12.4.1 Répertoire des événements multicast : `sdr`

Une fois qu'on est connecté au réseau MBone, on peut souhaiter se tenir au courant des événements proposés par les organismes extérieurs. De même, on peut souhaiter annoncer les informations qu'on veut rendre disponible. C'est le rôle de `sdr`.

`sdr` se présente sous la forme d'une liste telle que celle présentée figure 12.7 page suivante. Cette liste contient le nom des sessions consultables, avec une note supplémentaire pour indiquer que cette session est privée (c'est notamment le cas de certaines sessions organisées par Cisco qui utilise le MBone pour faire communiquer ses développeurs, mais qui ne souhaite pas que des personnes étrangères à la société participent à la conversation). Lorsqu'on choisit une session, `sdr` lance les programmes adéquats permettant de participer aux différents événements.

Le fait de choisir par exemple la session intitulée « FreeBSD Lounge » ouvre une fenêtre (montrée figure 12.8 page ci-contre) présentant une description de la session et les différents moyens de transmission proposés sur ce thème.

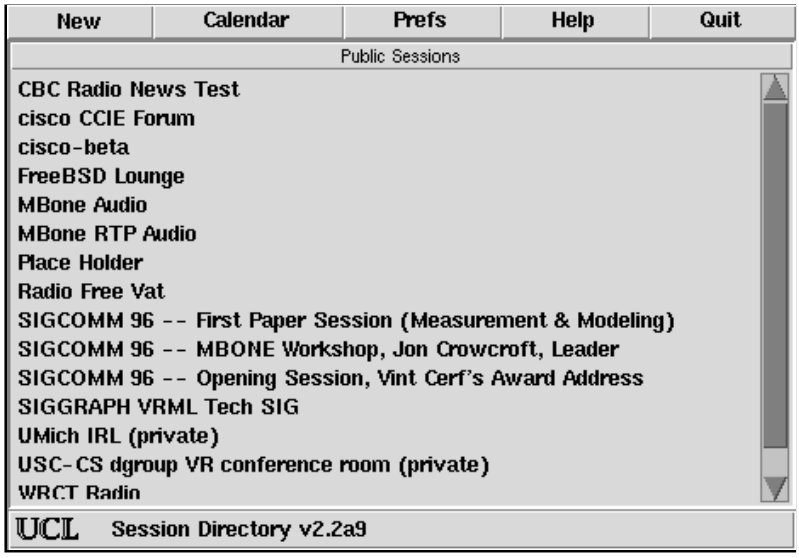

Figure 12.7 *Exemple de session sdr*

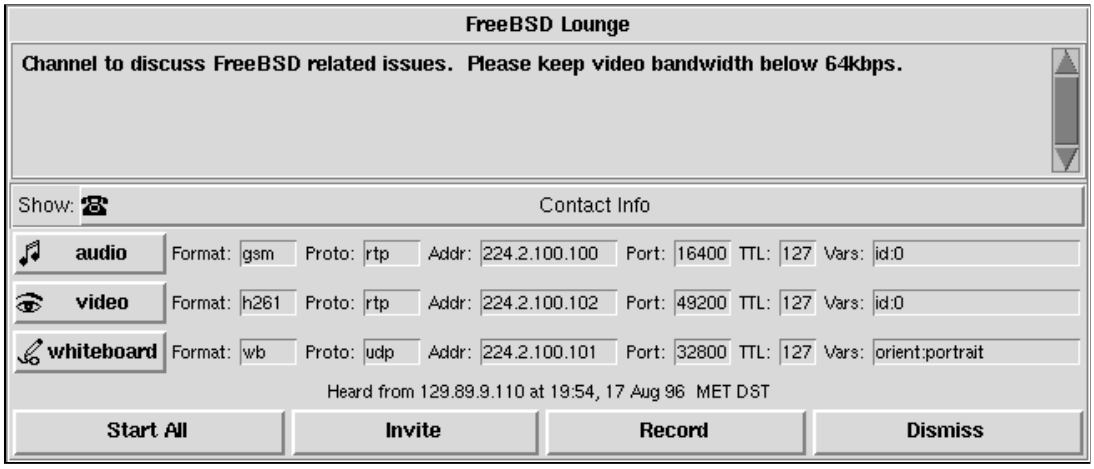

Figure 12.8 *Détails d'une session sdr*

Dans cet exemple, il apparaît que cette session est prévue pour discuter du système d'exploitation « FreeBSD » et une note prie les participants de ne pas transmettre de vidéo à plus de 64 kilobits par seconde. On voit également que trois média sont présentés, à savoir de l'audio, de la vidéo et un tableau blanc. Il est possible, à partir de cette fenêtre, de sélectionner outil par outil ce que l'on veut recevoir, ou de demander à lancer l'ensemble des outils possibles pour cette session.

La figure 12.9 page suivante montre la fenêtre qu'on obtient lorsqu'on choisit, à l'aide du bouton « New », de créer une nouvelle session à laquelle pourront s'abonner d'autres personnes. Les paramètres à définir sont :

Nom de la session : c'est le nom court qui apparaîtra dans la liste des sessions disponibles. Il doit être suffisament significatif pour ne pas induire en erreur les autres utilisateurs.

Il est conseillé de mettre le mot « test » dans le nom si la session n'est pas destinée à être sérieuse et maintenue.

Description : ce champ doit contenir, sur une ou plusieurs lignes, un descriptif de la session concernée. Si cette session est destinée à être diffusée en dehors du pays, cette description doit être idéalement en anglais.

URI : on peut mettre ici, mais cela n'a rien d'obligatoire, une adresse WWW décrivant la session ou l'organisme qui l'a initiée. Le bouton « Test URI » qui l'accompagne permet d'utiliser le client WWW intégré dans `sdr` afin de vérifier la disponibilité de la page.

Portée : ici, on a choisi de limiter la session au site. Il est possible de la limiter à la région ou au monde.

Media : une liste des médias que l'on souhaite utiliser est disponible. Dans cet exemple, seule la vidéo nous intéresse. Le format choisi est **H.261**, mais on aurait pu en choisir un autre tel que **nv** ou **JPEG**.

Calendrier prévisionnel : la session est annoncée ici à partir du dimanche 18 août à 15 heures 30 pour une durée de deux heures. On peut choisir d'annoncer une session régulière en utilisant le champ permettant d'indiquer la répétition.

Point de contact : ces champs doivent contenir les points de contacts (adresse électronique et numéro de téléphone). Ils seront utilisés pour contacter le créateur de la session en cas de problème de transmission ou de saturation de bande passante par exemple.

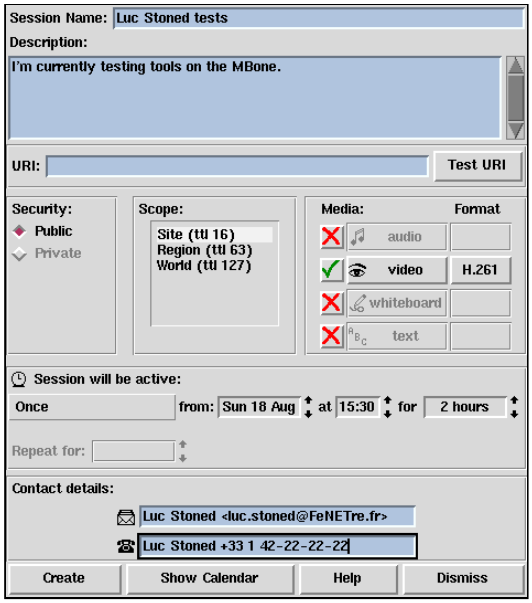

Figure 12.9 *Création d'une nouvelle session*

12.4.2 L'audio

La transmission du son a constitué la première phase du réseau MBone tel que nous le connaissons actuellement. Il existe actuellement deux outils largement répandus, **VAT** et **RAT**, classés par ordre d'antériorité.

VAT

VAT a été le premier outil de diffusion audio sur multicast à avoir été utilisé à grande échelle sur l'Internet. Dans une fenêtre VAT traditionnelle (telle que celle présentée figure 12.10), on peut lire la liste des participants à la session audio.

Lorsqu'un des participants prend la parole, la case se trouvant devant son nom se noircit alors qu'on l'entend parler.

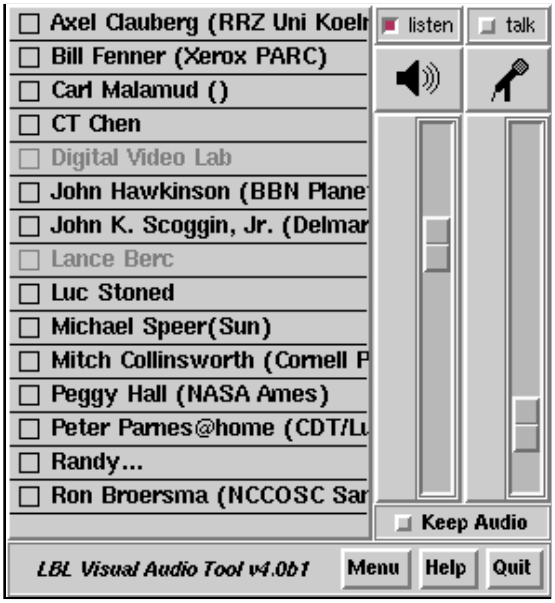

Figure 12.10 *Exemple de session vat*

Les noms qui apparaissent en grisé sont ceux des personnes qui sont actuellement inaccessibles : le programme VAT envoie régulièrement en multicast un signal indiquant que la personne annoncée est toujours là ; au bout de l'expiration d'un certain délai, une personne n'étant pas annoncée est considérée comme un «mort-vivant» : elle semble ne plus être là mais on ne l'a pas vue partir.

En cliquant sur le bouton « Menu », on accède à un panneau de configuration présenté figure 12.11 page suivante qui permet notamment de configurer les paramètres personnels (nom de l'utilisateur par exemple). D'ici on peut aussi choisir éventuellement une clé de session, utile lorsqu'on souhaite crypter les paquets afin de rendre la session privée et inaccessible à ceux qui ne disposent pas de ce mot de passe.

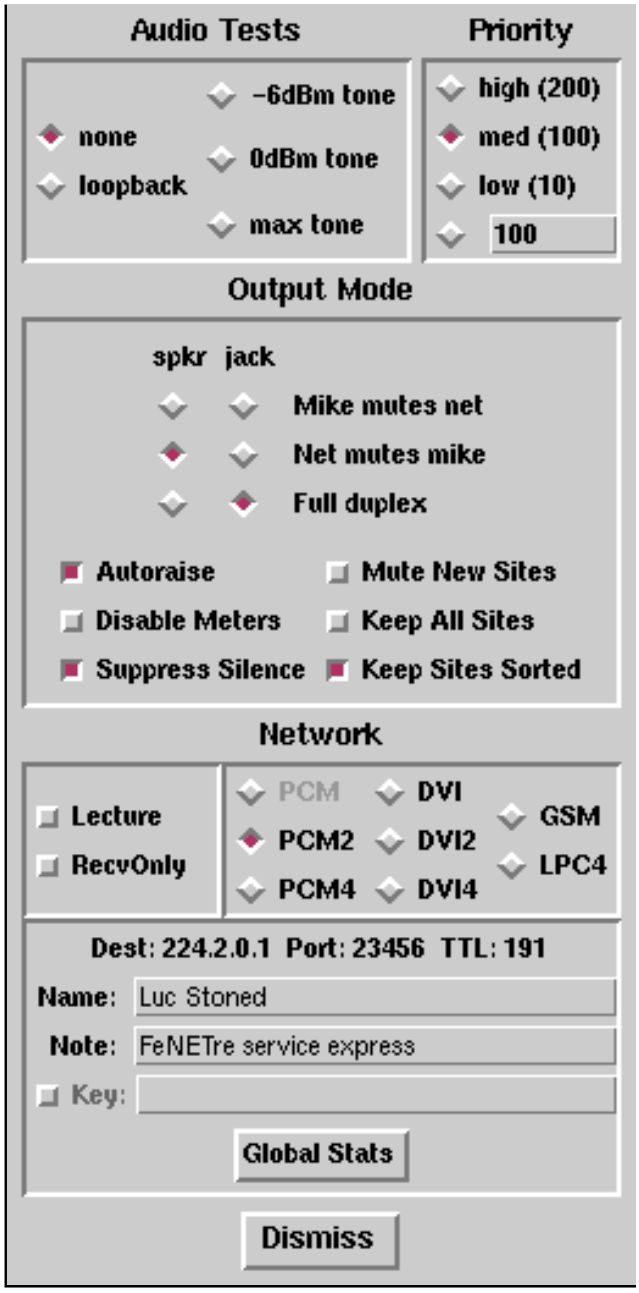

Figure 12.11 *Panneau de configuration*

RAT

L'utilisation de RAT (présentée figure 12.12 page ci-contre) est très semblable à celle de VAT. RAT a été écrit au département informatique du fameux *University College* de Londres. Il utilise une technique basée sur la redondance de l'information pour reconstruire les paquets perdus, ce qui le rend plus fiable en environnement perturbé. Il comporte en outre certaines options supplémentaires par rapport à VAT, mais est pour l'instant moins répandu sur l'Internet.

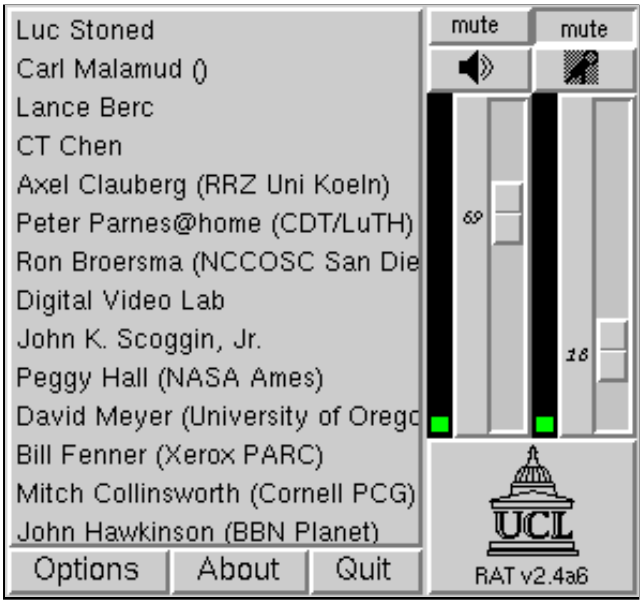

Figure 12.12 *Exemple de session rat*

12.4.3 La vidéo

VIC

VIC est certainement l'outil vidéo le plus présent à l'heure actuelle sur l'Internet. Un exemple de session VIC est présenté figure 12.13. On peut y voir une image réduite de la session en cours (il peut y avoir plusieurs images s'il y a plusieurs émetteurs), ainsi que des boutons permettant de choisir si on souhaite recevoir les images en couleur ou en noir et blanc, ou si on souhaite ne pas décoder les émissions vidéo de la session en cours (c'est notamment le cas lorsque l'on est le seul émetteur et que l'on ne souhaite pas avoir de « miroir » de notre propre émission).

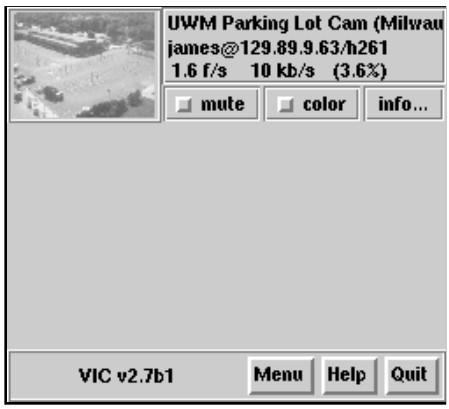

Figure 12.13 *Exemple de session vic (sommaire)*

En cliquant sur cette image, on ouvre une nouvelle fenêtre (figure 12.14 page suivante) de

taille plus confortable. On peut apercevoir sur cette copie d'écran la vidéo en temps réel d'un parking américain sur lequel est braqué une caméra, dès lors que les participants de la session « FreeBSD Lounge » ne sont pas actifs. Quelques boutons, dans le bas de l'image, permettent de choisir la taille de l'image ainsi que le mode de décodage du signal vidéo. On peut, bien évidemment, fermer cette fenêtre à l'aide du bouton « Dismiss ».

Figure 12.14 *Exemple de session VIC (image)*

Si le message « Waiting for video... » apparaît dans une fenêtre dans laquelle on attendait une image, cela signifie que personne ne transmet actuellement d'image dans le cadre de la session concernée. Dans ce cas, il nous faut soit attendre une transmission vidéo soit en lancer une nous-même.

Pour transmettre une image vidéo, telle que celle présentée sur la figure 12.15 page suivante, il faut disposer d'une caméra branchée sur une station de travail munie d'une carte d'acquisition, **SunVideo** par exemple. L'image, étant disponible en local, est habituellement d'une grande qualité.

Le contrôle des paramètres de transmission est présenté figure 12.16 page ci-contre. Certains de ces paramètres doivent être vérifiés avec soin, car ils peuvent affecter la bande passante du MBone tout entier :

Transmission : les deux boutons en haut à gauche permettent d'activer ou de désactiver la transmission, ainsi que de libérer le périphérique d'acquisition afin de le rendre disponible pour une autre application.

Contrôle du débit : les deux curseurs permettent de contrôler la bande passante utilisée en fixant un débit maximum qui sera employé pour la transmission de l'image ainsi que le nombre maximum d'images par seconde qui seront transmises. Lorsqu'on émet en dehors de l'entreprise (dans le pays par exemple), on doit vérifier que ces paramètres sont compatibles avec les autres émissions en cours, pour éviter d'engorger les liaisons.

Périphérique : un ou plusieurs périphériques sont sélectionnables, par exemple la carte Pa-

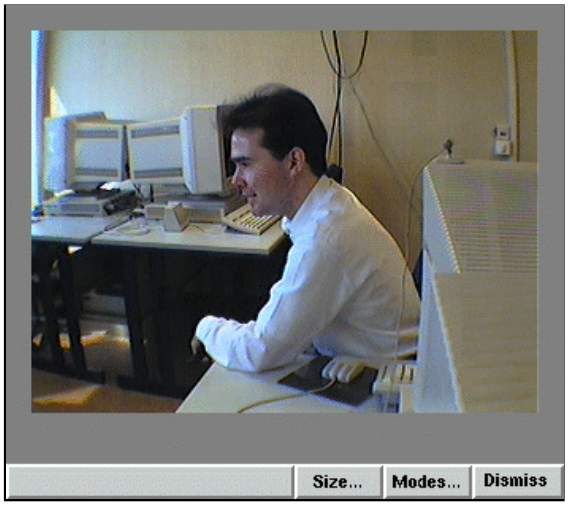

Figure 12.15 *Exemple de session VIC (image locale)*

rallax ou la carte SunVideo, ainsi qu'un ou plusieurs ports.

Format et qualité : on peut choisir le format dans lequel on souhaite réaliser les émissions vidéo, ainsi que la qualité de l'image (30 étant la meilleure et 1 la plus médiocre).

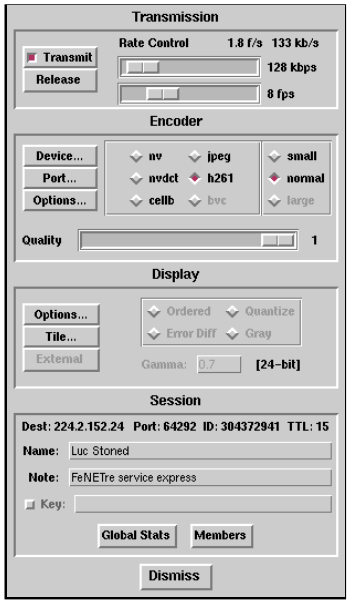

Figure 12.16 *Contrôle d'une session VIC*

QUATRIÈME PARTIE

Sécurité

Se raccorder en toute sécurité à l'Internet demande un minimum de précautions : la mise en place d'un *Firewall* protège efficacement contre d'éventuelles tentatives d'aggression.

13

Le firewall

Lorsqu'un professionnel se raccorde à l'Internet, il ouvre son réseau local à un réseau mondial. Même s'il se contente d'utiliser l'Internet pour ses services, il faut qu'il ait pleinement conscience que la porte ouverte qui lui permet d'accéder aux fabuleuses richesses du réseau permet aussi aux machines connectées sur l'Internet de rentrer chez lui. Pour se prémunir de ce type de mésaventure, la sécurité doit systématiquement faire partie intégrante de la problématique d'un raccordement au réseau.

13.1 Modèle du firewall

Un réseau connecté à l'Internet contient différents types de machines :

- les postes clients : ce sont les postes des utilisateurs ;
- les serveurs internes : ils sont destinés à fournir des services aux postes clients. Par exemple, un serveur FTP de transfert de fichiers entre les postes internes constitue un serveur interne ;
- les serveurs externes : ils sont destinés à fournir des services à l'Internet, cela englobe par exemple les serveurs WWW ou les serveurs DNS.

Les machines de notre réseau peuvent donc être classées en machines exposées et machines privées. En se basant sur cette remarque, on définit la topologie de notre réseau sécurisé, qui comprend deux sous-réseaux :

- Le réseau privé : il accueille les machines internes. Une série de filtres sur les différents routeurs interdit les connexions directes entre l'Internet et les machines de ce réseau.

– Le réseau bastion : il accueille les machines exposées, et joue le rôle de zone de démarcation. Les serveurs de ce réseau peuvent communiquer avec les machines du réseau privé et avec les machines de l'Internet.

La figure 13.1 présente le principe de ce type de topologie.

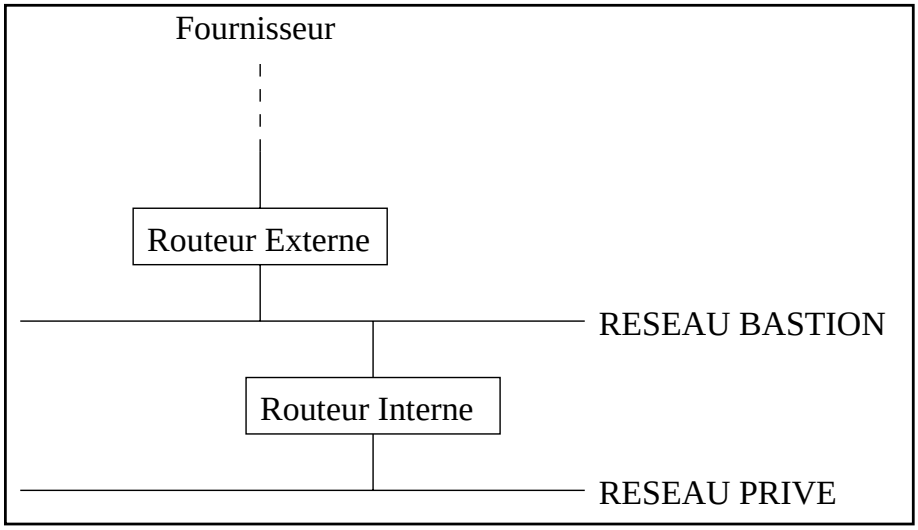

Figure 13.1 *Premier modèle firewall*

On peut constater sur cette figure que deux routeurs sont nécessaires. Cela augmente le nombre de filtres et leur complexité. De plus, pour passer du réseau privé à l'Internet, il faut systématiquement traverser le réseau bastion, qui, dans ce modèle, est le maillon faible car plus ouvert.

Un autre modèle à l'aide d'un routeur muni de trois interfaces est parfois proposé, comme on peut l'observer sur la figure 13.2. On constatera dans ce deuxième modèle qu'un seul routeur est nécessaire et qu'on n'est pas systématiquement obligé de traverser le réseau bastion lorsqu'on veut tranférer des informations entre le réseau privé et l'Internet.

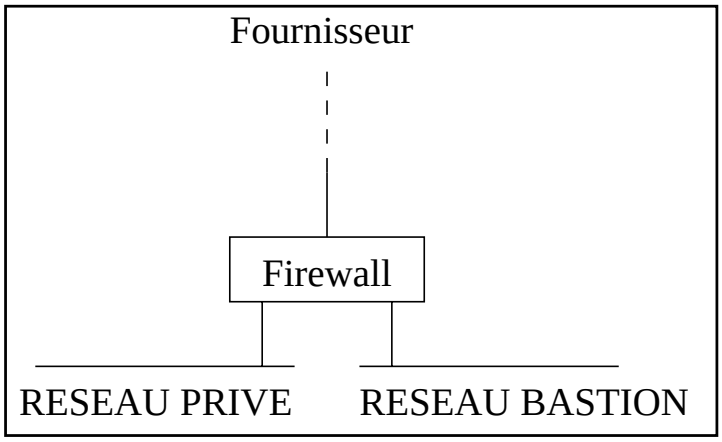

Figure 13.2 *Deuxième modèle firewall*

13.2 Filtrage

L'Internet est fondé sur les protocoles TCP/IP. C'est donc un réseau à commutation de paquets. Pour mettre en place des filtres pour se protéger des connexions indésirables, on doit donc configurer les routeurs pour qu'ils analysent les informations contenues dans chaque datagramme IP.

13.2.1 Principe

Un datagramme IP contient un en-tête qui fournit trois informations fondamentales pour un routeur filtrant :

- une adresse IP source : c'est l'adresse IP du nœud qui a émis ce datagramme ;
- une adresse IP destination : c'est l'adresse IP du nœud à qui ce datagramme est destiné ;
- le type de protocole encapsulé : ce paramètre indique le format des données encapsulées dans le datagramme. Il peut s'agir par exemple de TCP, UDP, ICMP ou IP. Les paquets TCP sont utilisés dans le cadre des services en mode connecté, les paquets UDP dans le cadre des services en mode non connecté. Les paquets ICMP permettent les opérations de gestion et de contrôle du réseau. Lorsqu'un datagramme IP encapsule le protocole IP, on parle alors de tunnel IP sur IP.

Lorsque le datagramme IP encapsule un paquet TCP ou UDP, le routeur filtrant sait analyser le début des données pour en extraire d'autres informations : il s'agit de l'en-tête du paquet. Elle contient deux paramètres fondamentaux :

- le numéro de port source : il s'agit d'un numéro choisi par l'émetteur du paquet, unique pour la session en question ;
- le numéro de port destination : c'est un numéro qui permet à la machine destinataire de déterminer à quel processus les données du paquet sont destinées.

L'attribution de ces numéros de port est du ressort du client et du serveur. Sur un serveur Internet, les différents services, comme WWW, FTP ou DNS, correspondent à des ports standard. Les serveurs WWW, FTP ou DNS, par exemple, sont des processus qui vont *écouter* sur le port local correspondant au service qu'ils gèrent. Les clients vont donc émettre des paquets TCP ou UDP à destination de ces ports connus de tous. Par exemple, le service WWW est localisé sur le port 80, le service DNS sur le port 23. Pour permettre aux serveurs de répondre aux requêtes des clients, ces derniers choisissent un port local, le plus souvent dynamiquement au moment du lancement de l'application.

Ainsi, chaque paquet émis par un client à destination d'un serveur possède comme port source celui choisi par le client et comme port destination celui défini par le protocole utilisé. Réciproquement, chaque paquet émis par un serveur à destination d'un client possède comme port source celui défini par le protocole et comme port destination celui choisi par le client.

Certains services nécessitent plusieurs connexions simultanées, c'est notamment le cas de FTP pour lequel les données transitent sur une liaison TCP distincte de celle dédiée aux commandes. Il y a donc plusieurs ports en jeu pour le même service. Le lecteur est prié de se reporter à la section 2.4.6 page 46 pour une liste plus complète des numéros de port standard.

On définit donc une règle de filtrage à partir de cinq informations :

1. adresse IP source,
2. adresse IP destination,
3. type de protocole encapsulé (TCP, UDP, ICMP, IP),
4. port source (au cas où le protocole est TCP ou UDP),
5. port destination (au cas où le protocole est TCP ou UDP).

Ces informations sont disséminées au sein des différentes en-têtes des paquets qui traversent le réseau, encapsulés les uns dans les autres.

Un filtre comprend un groupe de règles de filtrage appliquées à une interface. Définir une règle de filtrage consiste à définir une classe de datagrammes en indiquant les valeurs des cinq paramètres distingués précédemment qui font entrer le datagramme dans la classe. À chaque règle de filtrage, on va associer une action : par exemple laisser passer le datagramme, ou le détruire, ou encore le détruire en activant une procédure d'exception.

Le principe du routeur filtrant est le suivant : à chacune de ses interfaces est appliqué un filtre en entrée et en sortie. Ainsi, un datagramme qui traverse le routeur filtrant va passer successivement à travers deux filtres :

1. le filtre appliqué en entrée de l'interface sur laquelle il est arrivé ;
2. le filtre appliqué en sortie de l'interface sur laquelle il est réémis.

Pour réduire le nombre de filtres, on peut n'appliquer un filtre qu'aux sorties des différentes interfaces. Ainsi, les datagrammes ne sont filtrés qu'une fois : sur l'interface par laquelle ils quittent le routeur.

Notons bien que dans le cadre d'un quelconque service Internet, le dialogue est à double sens : les datagrammes vont donc traverser le routeur filtrant dans les deux sens, et pour chaque sens c'est le filtre de sortie d'une interface distincte qui rentrera en jeu. Mettre en place le filtrage d'un service consiste ainsi à mettre en place deux filtres sur chaque routeur.

Dans notre topologie, il y a deux routeurs. Il faut donc, pour chaque service, définir le ou les routeurs qui peuvent être traversés par les datagrammes du service afin de définir les différents filtres à mettre en place.

Par défaut, le filtre ne laisse rien passer. Si on a oublié de mettre en place la règle de filtrage d'un service donné, ce service est simplement indisponible, mais aucun problème de sécurité ne peut en résulter.

13.2.2 Filtrage UDP

Principe

Etudions maintenant le filtrage d'un service UDP. Nous voulons laisser passer un service UDP à travers un de nos deux routeurs filtrants. Il faut tout d'abord se renseigner sur le port du service en question. Désignons par A une machine désireuse d'accéder à ce service et par B la machine qui héberge le serveur. Désignons par Pb le numéro de port du service en question. Ce numéro est imposé par le protocole du service correspondant. La machine cliente A choisit un port source Pa. Les deux types de paquets susceptibles de traverser notre routeur sont indiqués sur la figure 13.3 :

– paquets UDP de A vers B dont le port source est Pa et le port destination est Pb. Ces paquets vont être filtrés sur l'interface du côté B, c'est-à-dire sur l'interface Ethernet 0 de notre exemple ;

– paquets UDP de B vers A dont le port source est Pb et le port destination est Pa. Ces paquets vont être filtrés sur l'interface du côté A, c'est-à-dire sur l'interface Ethernet 1 de notre exemple.

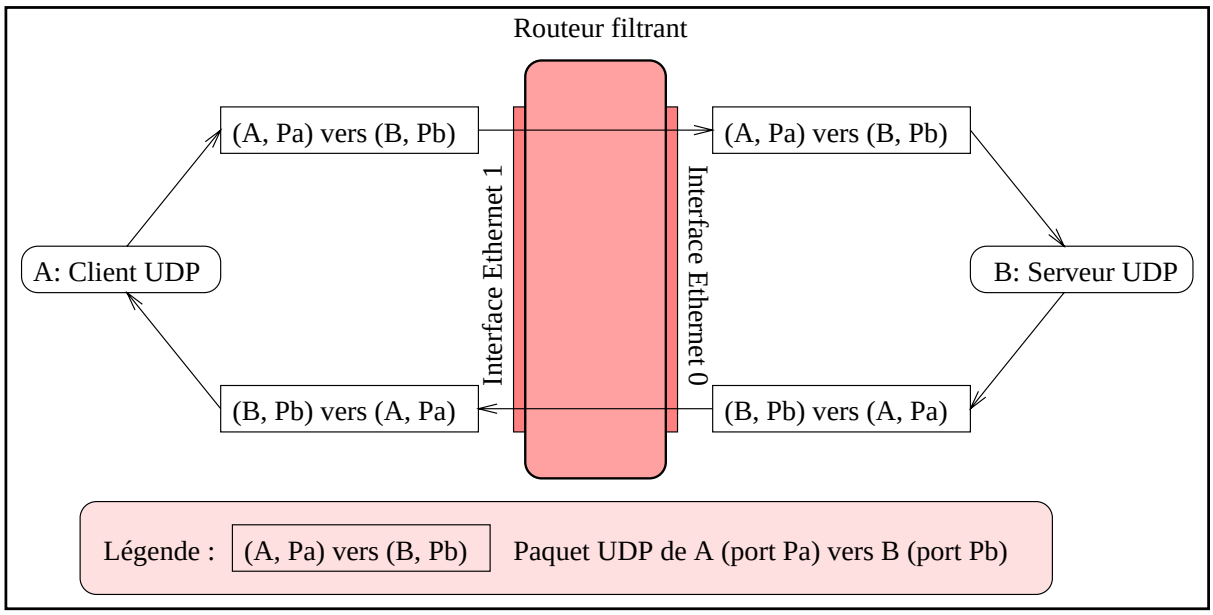

Figure 13.3 *Filtrage d'un service UDP*

On va donc définir les plages d'adresses IP de A et B, et les plages de ports Pa et Pb.

On configure alors les filtres des interfaces Ethernet de notre routeur filtrant. On peut indiquer les adresses IP de A et B en fournissant des numéros particuliers, ou des réseaux de classe A, B ou C, afin de n'autoriser que certaines machines à dialoguer par UDP avec le protocole en question.

Pb est une valeur connue. Souvent, Pa est choisi au hasard. C'est par exemple le cas des accès au DNS par l'utilitaire `dig`. Ce n'est par contre pas le cas lorsqu'un DNS répond à un autre DNS.

On voit ainsi que, pour chaque service, par deux règles particulières de filtrage, on autorise les paquets à traverser un routeur *dans un sens bien défini* : les règles concernant les adresses source et destination, ainsi que les ports sont différentes qu'on veuille autoriser les machines de notre réseau bastion à accéder à des serveurs sur l'Internet, ou qu'on veuille autoriser l'Internet à accéder à des serveurs sur notre réseau bastion. Ainsi, pour chaque service, il faut mettre en place des règles pour autoriser nos logiciels clients à y accéder sur l'Internet, et des règles pour autoriser des applications clientes sur l'Internet à accéder à des serveurs sur notre réseau local.

Autoriser l'extérieur à accéder à un service sur le réseau bastion

Prenons un exemple particulier : le filtrage du service DNS.

Sur notre réseau bastion, nous disposons d'un serveur DNS primaire d'une zone particulière. Son adresse IP est `192.168.22.35`. Nous désirons que ce serveur puisse recevoir des requêtes depuis d'autres serveurs DNS sur l'Internet, et même depuis des clients du service DNS, toujours sur l'Internet. Nous savons que le serveur écoute sur le port UDP 53. Les clients sur l'Internet vont choisir un port source quelconque. Les filtres correspondants, appliqués en sortie des interfaces de réseau local et distant sur le routeur raccordant le réseau local au fournisseur, sont donc :

Règle	**IP source**	**IP destination**	**Port source**	**Port destination**	**Action**
WAN-1	`192.168.22.35`	quelconque	UDP/53	quelconque	laisser sortir
WAN-2	quelconque	quelconque	quelconque	quelconque	détruire
LAN-1	quelconque	`192.168.22.35`	quelconque	UDP/53	laisser sortir
LAN-2	quelconque	quelconque	quelconque	quelconque	détruire

Les règles WAN-1 et WAN-2 sont les deux règles appliquées en sortie sur l'interface vers le fournisseur. Les règles LAN-1 et LAN-2 sont les deux règles appliquées en sortie vers le réseau local sur l'interface Ethernet. Sur une même interface, les règles sont évaluées dans l'ordre, et dès qu'une d'entre elles correspond au paquet en cours de traitement, l'action correspondante est entreprise et les règles qui suivent sont ignorées. Ainsi, les règles WAN-2 et LAN-2 permettent de refuser les paquets qui ne correspondent à aucune des autres règles.

Autoriser le réseau bastion à accéder à un service extérieur

Reprenons notre exemple : le filtrage du service DNS.

Nous désirons que les machines du réseau bastion puissent faire des requêtes DNS sur l'Internet. L'application cliente est interne, et choisit un port source quelconque. L'application distante sur l'Internet est un serveur DNS : le numéro de port est 53.

Les filtres sont donc les suivants :

Règle	**IP source**	**IP destination**	**Port** source	**Port** destination	**Action**
WAN-1	`192.168.22/24`	quelconque	quelconque	UDP/53	laisser sortir
WAN-2	quelconque	quelconque	quelconque	quelconque	détruire
LAN-1	quelconque	`192.168.22/24`	UDP/53	quelconque	laisser sortir
LAN-2	quelconque	quelconque	quelconque	quelconque	détruire

La règle LAN-1 pose un problème important et général dans le cadre de l'accès à un service UDP extérieur : elle permet à n'importe qui sur l'Internet d'émettre sur notre réseau local des paquets UDP vers un port quelconque, donc vers un service quelconque.

Deux solutions peuvent être apportées à ce problème :

– Si le service UDP auquel on veut accéder est disponible sur une machine particulière dont on connaît l'adresse IP, on peut préciser les règles de filtrage avec cette adresse. Par exemple, il peut s'agir d'une machine sur laquelle on veut monter par `NFS` un système de fichiers particulier.

  Ainsi, seule la machine en question, dans laquelle il faut avoir confiance, ne peut émettre des paquets indésirables sur notre réseau.

– L'autre solution consiste à examiner plus précisément le cas particulier du protocole qui nous intéresse ici : le DNS. Cette solution par cas particulier n'est pas toujours envisageable avec n'importe quel protocole.

  Dans le cadre du DNS, on peut mettre en place un serveur DNS sur le site plutôt que d'utiliser celui du fournisseur pour résoudre les requêtes. On configure nos clients DNS pour attaquer ce serveur et non directement l'Internet. Le serveur va retransmettre les requêtes DNS en choisissant, quant à lui, un port source connu, le port 53.

Examinons les règles de filtrage dans le cadre de la présence d'un serveur DNS :

Règle	**IP source**	**IP destination**	**Port source**	**Port destination**	**Action**
WAN-1	`192.168.22.35`	quelconque	UDP/53	UDP/53	laisser sortir
WAN-2	quelconque	quelconque	quelconque	quelconque	détruire
LAN-1	quelconque	`192.168.22.35`	UDP/53	UDP/53	laisser sortir
LAN-2	quelconque	quelconque	quelconque	quelconque	détruire

La règle qui nous posait problème a donc disparu.

13.2.3 Filtrage TCP

Principe

Etudions maintenant le filtrage d'un service TCP. Nous voulons laisser passer un service TCP à travers un de nos deux routeurs filtrants. Il faut tout d'abord se renseigner sur le port du service en question. De même que lors de l'étude du filtrage UDP, désignons par A une machine désireuse d'accéder à ce service et par B la machine qui héberge le serveur. Désignons par Pb le port du service en question. Ce nombre est imposé par le protocole du service correspondant. La machine cliente A choisit un port source Pa. Les trois types de paquets susceptibles de traverser notre routeur sont indiqués sur la figure 13.4 :

- premier paquet TCP de la connexion : il part de A pour aller vers B, il est filtré sur l'interface du côté B, c'est-à-dire sur l'interface Ethernet 0 de notre exemple ;
- paquets TCP de A vers B dont le port source est Pa et le port destination est Pb, et distincts du premier de la connexion. Ces paquets vont être filtrés sur l'interface du côté B ;
- paquets TCP de B vers A dont le port source est Pb et le port destination est Pa. Ces paquets vont être filtrés sur l'interface du côté A, c'est-à-dire sur l'interface Ethernet 1 de notre exemple.

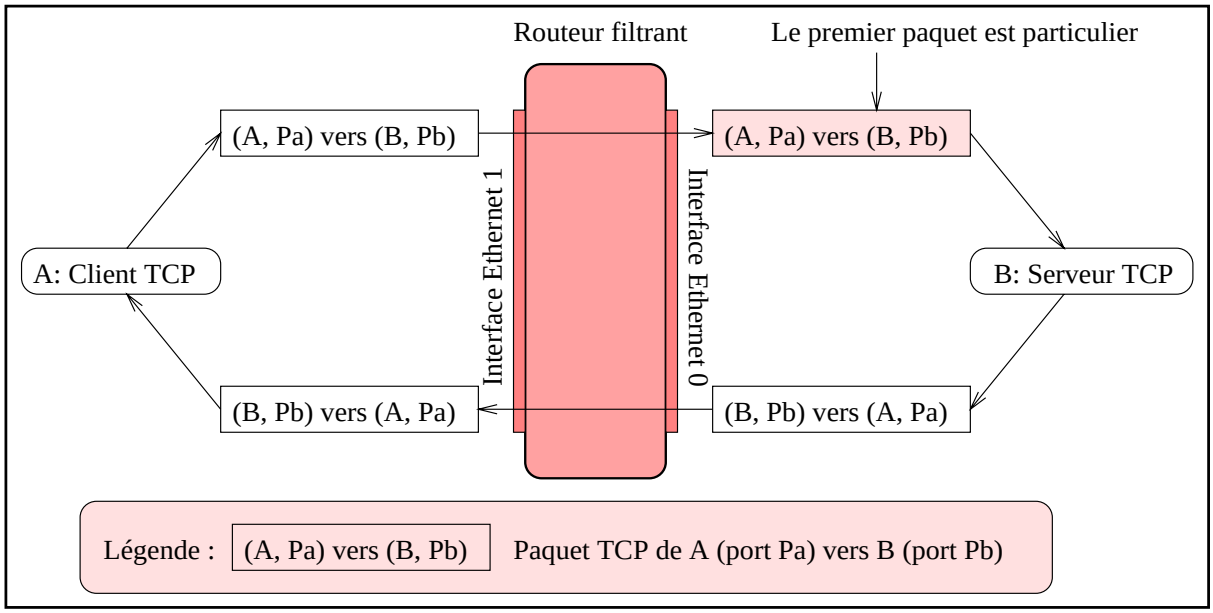

Figure 13.4 *Filtrage d'un service TCP*

On remarque qu'avec TCP, le premier paquet de la connexion est particulier. Plus précisément, les deux premiers paquets peuvent être distingués mais la connaissance du premier suffit pour les filtres. Cette information supplémentaire que nous n'avons pas dans le cadre des échanges UDP va nous permettre de résoudre le problème de l'accès à un service externe

de manière générale. Rappelons que nous l'avions traité précédemment au cas par cas pour UDP.

Autoriser l'extérieur à accéder à un service sur le réseau bastion

Prenons un exemple particulier : nous voulons proposer un service WWW hébergé sur notre serveur d'adresse IP `192.168.22.35`. Tout l'Internet doit donc pouvoir y accéder. Notre serveur WWW écoute sur le port du protocole HTTP qui porte le numéro 80. Les clients de notre serveur, par exemple des navigateurs de type Netscape ou Internet Explorer, vont ainsi initier des connexions TCP depuis un port quelconque vers le port 80 de la machine qui héberge notre serveur. On ajoute une colonne pour décrire les filtres TCP : elle indique un drapeau qui peut être ou ne pas être présent sur la règle en question : il s'agit du drapeau `established` qui indique que la règle ne s'applique pas au premier paquet d'une connexion TCP. Les filtres correspondants, appliqués en sortie des interfaces de réseau local et distant sur le routeur raccordant le réseau local au fournisseur, sont donc :

Règle	IP source	IP destination	Port source	Port destination	drapeau	Action
WAN-1	`192.168.22.35`	quelconque	TCP/80	quelconque	establ.	laisser sortir
WAN-2	quelconque	quelconque	quelconque	quelconque	/	détruire
LAN-1	quelconque	`192.168.22.35`	quelconque	TCP/80	/	laisser sortir
LAN-2	quelconque	quelconque	quelconque	quelconque	/	détruire

Autoriser le réseau bastion à accéder à un service extérieur

Reprenons notre exemple : le filtrage du protocole HTTP. Nous désirons que les machines du réseau bastion puissent faire des requêtes HTTP pour accéder à des serveurs WWW sur l'Internet. L'application cliente choisit un port source quelconque. L'application distante sur l'Internet est un serveur HTTP : le numéro de port destination est donc 80. Construisons donc les filtres correspondant :

Règle	IP source	IP destination	Port source	Port destination	drapeau	Action
WAN-1	`192.168.22.0/24`	quelconque	quelconque	TCP/80	/	laisser sortir
WAN-2	quelconque	quelconque	quelconque	quelconque	/	détruire
LAN-1	quelconque	`192.168.22.0/24`	TCP/80	quelconque	establ.	laisser sortir
LAN-2	quelconque	quelconque	quelconque	quelconque	/	détruire

La règle LAN-1 qui posait problème lors de ce type d'accès avec le protocole UDP ne pose pas de problème ici avec TCP : en effet, elle ne permet pas à n'importe qui sur l'Internet d'émettre des paquets sur notre réseau vers un port quelconque, donc sur n'importe quel service local, car le premier paquet d'une connexion TCP n'est pas autorisé par cette règle. Ainsi, le problème étudié avec UDP dans le cadre de l'accès à des services distants disparait avec TCP.

13.2.4 Protection contre l'IP-spoofing

La technique d'IP-spoofing est un type d'attaque bien connu sur l'Internet. Elle consiste à tromper les vérifications de certains logiciels en émettant des datagrammes à destination d'un réseau cible dont les adresses IP source ont été falsifiées et choisies dans les plages d'adresses attribuées au réseau cible lui-même.

Certains logiciels, comme les *TCP-Wrappers* que nous étudierons section 13.5 page 416, permettent de bloquer les accès provenant de l'extérieur du réseau. On peut ainsi, par exemple, filtrer les connexions `telnet` sur une machine Unix en mettant en place un TCP-Wrapper qui va limiter les accès en examinant les adresses sources des connexions qui sont établies avec lui. Il décide alors d'activer le démon `in.telnetd` ou non en fonction de ces adresses. Il s'agit ici d'un Wrapper et non d'un filtre : on substitue au démon `telnet` un autre programme, plutôt que de mettre en place des filtres sur un routeur. Ces deux techniques sont complémentaires et souvent utilisées de pair.

L'IP-spoofing est ainsi particulièrement adapté pour tromper par exemple un TCP-Wrapper. On prend donc systématiquement la précaution de mettre en place un filtre particulier destiné à se protéger contre ce type d'attaque. Ce filtre va détruire les datagrammes à destination du réseau local dont les adresses source font aussi partie :

Règle	**IP source**	**IP destination**	**Port source**	**Port destination**	**Action**
LAN-1	`192.168.22.0/24`	quelconque	quelconque	quelconque	détruire

13.2.5 Mise en place d'un filtre sur un routeur dédié

Une fois la politique de filtrage définie et les règles de filtrage écrites, il faut les transcrire dans le langage propre au routeur filtrant qu'on utilise. Ces langages dépendent du constructeur du routeur. Si on utilise une machine Unix équipée du logiciel MorningStar PPP, la syntaxe sera différente de celle utilisée avec un routeur CISCO.

Etudions maintenant la syntaxe en vigueur avec les routeurs CISCO, qui sont très utilisés en tant que routeurs filtrants, munis de deux cartes Ethernet et placés entre le réseau bastion et le réseau privé, et en tant que routeurs de proximité entre le réseau bastion et le réseau du fournisseur, comme on a pu le constater lors de l'étude de la connexion du réseau local à l'Internet.

À chaque filtre, ou *access-list*, est associé un numéro. Si ce numéro est compris entre 100 et 199, le filtre est de type *extended access-list*, ou liste d'accès étendue. La syntaxe des listes d'accès étendues va nous permettre de définir simplement les adresses IP et numéros de port du filtre qu'on veut mettre en place.

Depuis la version 11.0 du système d'exploitation IOS des matériels CISCO, la commande `access-list` propose de nombreux paramètres :

– la commande se conforme à la syntaxe générale que voici :

```
access-list access-list-number {deny | permit} protocol \
                                  source source-wildcard \
                         destination destination-wildcard \
                [precedence precedence] [tos tos] [log]
```

– pour les paquets ICMP, on peut aussi utiliser la syntaxe plus complète suivante :

```
access-list access-list-number {deny | permit} icmp \
                 source source-wildcard destination \
                                destination-wildcard \
             [icmp-type [icmp-code] | icmp-message] \
            [precedence precedence] [tos tos] [log]
```

– pour les paquets IGMP, on peut aussi utiliser la syntaxe plus complète suivante :

```
access-list access-list-number {deny | permit} igmp \
                               source source-wildcard \
                   destination destination-wildcard \
               [igmp-type] [precedence precedence] \
                                    [tos tos] [log]
```

– pour les paquets TCP, on peut aussi utiliser la syntaxe plus complète suivante :

```
access-list access-list-number {deny | permit} tcp       \
           source source-wildcard [operator port [port]] \
 destination destination-wildcard [operator port [port]] \
   [established] [precedence precedence] [tos tos] [log] \
```

– pour les paquets UDP, on peut aussi utiliser la syntaxe plus complète suivante :

```
access-list access-list-number {deny | permit} udp       \
           source source-wildcard [operator port [port]] \
 destination destination-wildcard [operator port [port]] \
                 [precedence precedence] [tos tos] [log]
```

Examinons les différents paramètres :

– `access-list-number` : le numéro du filtre. Il doit être le même pour toutes les règles du même filtre. Ces dernières seront parcourues dans l'ordre où elles auront été saisies ;
– `deny` : interdire l'accès si les conditions sont réunies ;
– `permit` : autoriser l'accès si les conditions sont réunies ;
– `protocol` : il s'agit du protocole encapsulé dans le datagramme IP. Les valeurs autorisées sont : `eigrp`, `icmp`, `igmp`, `igrp`, `gre`, `ip`, `ipinip`, `nos`, `ospf`, `tcp`, `udp` ou un numéro de protocole entre 0 et 255 ;
– `source` : adresse source ;
– `source-wildcard` : masque à appliquer à `source` ;
– `destination` : adresse destination ;
– `destination-wildcard` : masque à appliquer à `destination` ;

- `precedence` : les datagrammes peuvent être filtrés par importance en indiquant un niveau d'importance contenu dans leur en-tête. Il peut s'agir de `critical`, `flash`, `flash-override`, `immediate`, `internet`, `network`, `priority`, `routine` ;
- `tos (Type of Service)` : les datagrammes peuvent être filtrés par type de service. Il peut s'agir de `max-reliability, max-throughput, min-delay, min-monetary-cost, normal` ;
- `icmp-type` : type de message ICMP lorsqu'on filtre un paquet ICMP. Il s'agit d'un nombre compris entre 0 et 255 ;
- `icmp-code` : les paquets ICMP filtrés par type de message peuvent l'être aussi par le code ICMP optionnel compris entre 0 et 255 ;
- `icmp-message` : une chaîne de caractères peut être fournie pour préciser le type et éventuellement le code d'un message ICMP particulier ;
- `igmp-type` : type de paquet IGMP ;
- `operator` : afin de filtrer par numéro ou plage de ports source et destination, on utilise `operator` suivi d'un ou deux paramètres pour désigner le ou les ports concernés :
  - `lt` : le numéro du port destination du paquet doit être inférieur à celui fourni en paramètre ;
  - `gt` : le numéro du port destination du paquet doit être supérieur à celui fourni en paramètre ;
  - `eq` : le numéro du port destination du paquet doit être le même que celui fourni en paramètre ;
  - `neq` : le numéro du port destination du paquet doit être différent de celui fourni en paramètre ;
  - `range` : les deux paramètres qui suivent définissent une plage de ports.
- `established` : ce paramètre facultatif, lorsqu'il est présent, indique que cette entrée ne s'applique pas au premier paquet d'une connexion TCP. Sa signification est la même que celle du drapeau `established` qu'on a utilisé lors de la section sur la mise en place d'un filtre TCP ;
- `log` : permet de signaler le passage d'un message qui correspond à cette règle de filtrage.

Notons que certains de ces paramètres ne sont pas disponibles dans les versions antérieures à la version 11.0 d'IOS, notamment en ce qui concerne le filtrage par le port source des paquets TCP ou UDP. Il est donc extrêmement conseillé de disposer d'une version récente de ce système d'exploitation lorsqu'on veut mettre en place des filtres sur un routeur CISCO.

Après avoir mis en place un filtre, il faut l'appliquer à une interface, en entrée ou en sortie. Pour cela, on utilise la sous-commande d'interface `ip access-group` dont la syntaxe est la suivante :

```
ip access-group access-list-number [in | out]
```

Le paramètre `access-list-number` désigne le numéro de la liste d'accès étendue définie précédemment, et les paramètres `in` et `out` indiquent si ce filtre s'applique en entrée ou en sortie.

Prenons pour exemple le filtrage de l'accès au service WWW sur le routeur qui sépare le réseau bastion du réseau du fournisseur. Nous désirons ainsi que le serveur d'adresse IP `192.168.22.35` qui dispose d'un serveur proxy-cache WWW puisse établir des connexions HTTP vers l'Internet. L'interface côté réseau local s'appelle Ethernet 0 et l'interface côté réseau étendu s'appelle BRI 0 (accès Internet par RNIS). La configuration du routeur CISCO est donc la suivante :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#access-list 100 permit tcp 192.168.22.35 0.0.0.0 0.0.0.0 255.255.255.255 \
                                                                              eq 80
Router(config)#access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
Router(config)#access-list 101 permit tcp 0.0.0.0 255.255.255.255 192.168.22.35 0.0.0.0 \
                                                                           established
Router(config)#access-list 101 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
Router(config)#interface BRI 0
Router(config-if)#ip access-group 100 out
Router(config-if)#exit
Router(config)#interface ethernet 0
Router(config-if)#ip access-group 101 out
Router(config-if)#^Z
Router#
```

Il est possible de simplifier l'écriture à l'aide des raccourcis suivants :

– la plupart des ports des services standard peuvent être désignés directement par le nom du service plutôt que par le numéro du port. Ainsi, 80 sera remplacé par `www` ;

– les paramètres d'adresse et de masque désignant n'importe quelle machine « `0.0.0.0 255.255.255.255` » peuvent être remplacés par `any` ;

– les paramètres d'adresse et de masque désignant une machine particulière, par exemple « `192.168.22.35 0.0.0.0` », peuvent être remplacés par l'écriture suivante : `host 192.168.22.35`.

On obtient ainsi la configuration suivante :

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#access-list 100 permit tcp host 192.168.22.35 any eq www
Router(config)#access-list 100 deny ip any any
Router(config)#access-list 101 permit tcp any host 192.168.22.35 established
Router(config)#access-list 101 deny ip any any
Router(config)#interface BRI 0
Router(config-if)#ip access-group 100 out
Router(config-if)#exit
Router(config)#interface ethernet 0
Router(config-if)#ip access-group 101 out
Router(config-if)#^Z
Router#
```

13.3 Choix des adresses du réseau privé

Les adresses IP du réseau bastion doivent être routées sur l'Internet. Il faut donc attribuer aux machines de ce réseau les adresses indiquées par le fournisseur. Par contre, les adresses du réseau privé n'ont aucune raison d'être routées sur l'Internet car, justement, la politique de filtrage sur les deux routeurs du modèle firewall consiste à interdire les connexions directes entre l'Internet et le réseau privé, afin d'imposer l'utilisation de serveurs *proxy* disposés sur le réseau bastion. Ainsi, pour une sécurité maximale, on peut choisir des adresses IP sur le réseau bastion qui ne sont pas routées sur l'Internet. Même une erreur dans les filtres ne permettrait pas, ainsi, à un éventuel attaquant d'atteindre le réseau privé depuis l'Internet.

Des plages d'adresses particulières ont été prévues à cet effet, dans le RFC 1918. Il s'agit des réseaux suivants :

- le réseau de classe A `10.0.0.0` ;
- les 16 réseaux de classe B compris entre `172.16.0.0` et `172.31.0.0` ;
- les 256 réseaux de classe C compris entre `192.168.0.0` et `192.168.255.0`.

13.4 Mise en place des services

Pour compléter notre étude du modèle firewall, il nous faut maintenant disposer les services sur le réseau bastion et le réseau privé, comme indiqué sur la figure 13.5 page suivante. Examinons donc en détail les différents services correspondants, leur disposition sur le réseau, ainsi que les connexions TCP et UDP qu'ils mettent en jeu. Le lecteur désireux de configurer un routeur filtrant en déduira les règles de filtrage TCP et UDP correspondantes en s'inspirant des différents exemples qu'on a donnés précédemment.

13.4.1 Courrier électronique

Le protocole utilisé pour le transport du courrier électronique s'appelle SMTP. Il correspond à des connexions TCP vers le port 25, sur lequel le serveur SMTP est en attente.

Pour ce service, deux passerelles SMTP sont nécessaires :

- La passerelle du réseau privé : elle traite les courriers internes, et lorsqu'un message est destiné à l'Internet, elle le redirige vers la passerelle du réseau bastion. Elle accepte de plus les connexions SMTP provenant de la passerelle du réseau bastion : il s'agit des messages entrants.
- La passerelle du réseau bastion : elle fait office de relais entre les serveurs SMTP de l'Internet et le serveur du réseau privé. Pour cela, on doit l'autoriser à recevoir des

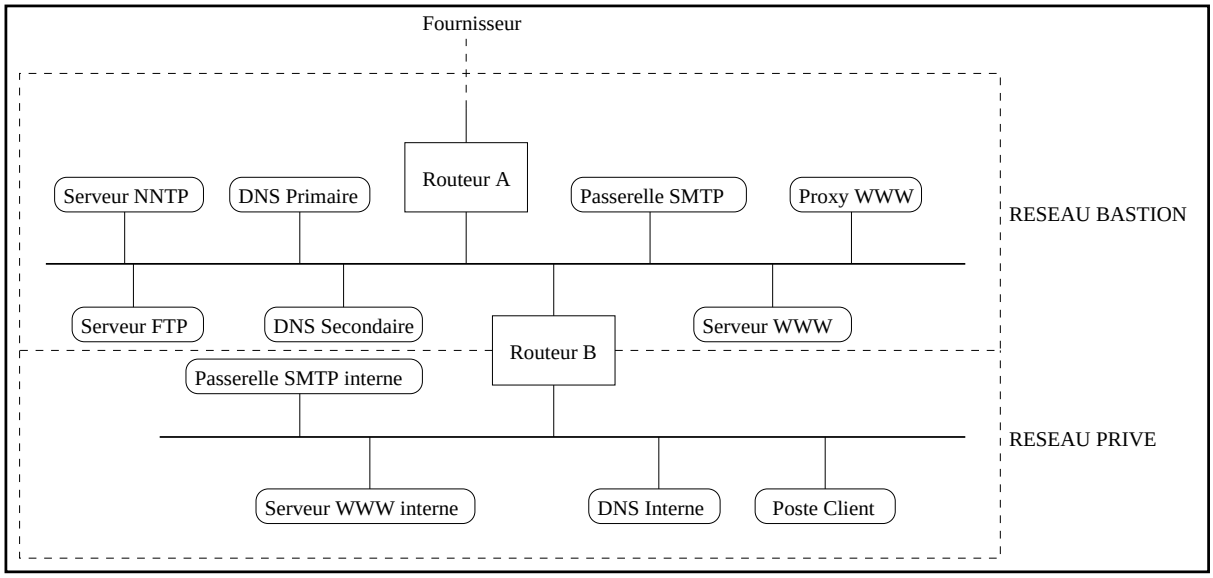

Figure 13.5 *Modèle firewall*

connexions sur le port 25 depuis l'Internet et depuis le serveur SMTP privé, et à créer des connexions vers le port 25 du serveur du réseau privé et vers l'Internet.

13.4.2 DNS

Le service DNS correspond à deux types d'échanges :

- des échanges UDP entre serveurs DNS dont les ports source et destination ont pour valeur 53 ;
- des connexions TCP de port destination 53, établies par les secondaires vers les primaires pour rapatrier les zones.

On dispose ainsi les serveurs primaire et secondaire sur le réseau bastion.

Sur le primaire, on propose le moins d'informations possible. On ne met par exemple pas d'informations sur les machines du réseau privé.

Sur le réseau privé, on dispose un serveur qui joue aussi le rôle de primaire, et qui contient l'ensemble des informations DNS du site. Il possède une clause *forwarder* vers les serveurs du réseau bastion afin de rediriger les requêtes non résolues en local vers l'Internet, et une clause *slave* pour imposer l'utilisation exclusive et systématique du *forwarder*.

13.4.3 Serveur WWW

Le serveur WWW destiné aux connexions depuis l'Internet est placé sur le réseau bastion. Le protocole HTTP utilisé ici correspond à des connexions TCP vers le port 80.

13.4.4 Serveur proxy

Le serveur *proxy* effectue des connexions TCP à destination de l'Internet vers des ports correspondant aux services dont il assure le relais (WWW, gopher, etc.).

Sachant qu'il existe de nombreux serveurs sur l'Internet dont le port n'est pas standard, il faut autoriser la machine *proxy* du réseau bastion à se connecter par TCP sur l'Internet vers n'importe quel numéro de port. Par exemple, on trouve beaucoup de serveurs WWW sur le port 8080 au lieu du port 80.

Pour l'accès au *proxy*, il faut autoriser les connexions depuis le réseau privé vers le port du *proxy* sur la machine qui l'héberge. Il s'agit habituellement du port 3128.

13.4.5 Forums

Le protocole de transport des forums s'appelle NNTP et son port porte traditionnellement le numéro 119.

Pour la consultation, il faut ainsi autoriser le serveur NNTP du fournisseur ainsi que les machines du réseau privé à se connecter sur le port 119 du serveur NNTP disposé sur le réseau bastion.

Pour l'émission, il faut de plus autoriser le serveur NNTP du réseau bastion à se connecter à celui du fournisseur.

Parfois, on conseille de disposer le serveur NNTP directement sur le réseau privé. Ce n'est néanmoins pas possible lorsque les adresses IP de ce réseau ont été choisies conformément au RFC 1918.

D'autres modèles proposent de chaîner deux serveurs NNTP : l'un sur le réseau bastion, l'autre sur le réseau privé. Celui du réseau bastion est un simple relais, celui du réseau privé permet la consultation.

L'idée sous-jacente est de ne pas faire transiter les données liées à la consultation sur le réseau bastion, ce qui est une règle générale dans le modèle du firewall, quel que soit le service. Mais il faut noter que l'administration de deux serveurs NNTP sur un même site est un travail important.

13.4.6 FTP

Le service FTP d'échange de fichiers est complexe à filtrer. Supposons que la machine A désire accéder au serveur FTP hébergé sur la machine B, deux connexions TCP sont alors mises en jeu :

– une connexion TCP de A vers B, vers le port destination `ftp` de numéro 21 et de port source quelconque ;

– une connexion TCP de B vers A, depuis le port source `ftp-data` de numéro 20 et de port destination quelconque, supérieur à 1024, attribué par le client, c'est-à-dire A.

Accès à des serveurs FTP sur l'Internet

Le premier problème qui se pose est d'établir les filtres pour l'accès à des serveurs FTP sur l'Internet. Pour cela, on dispose un *proxy* FTP sur le réseau bastion. On autorise donc sur le routeur B les connexions TCP vers le port *proxy* de numéro 3128 depuis les machines du réseau privé (la connexion *proxy* pour FTP est classique et ne présente aucune difficulté à filtrer).

Il faut maintenant autoriser le *proxy* à se connecter par FTP sur des serveurs distants. On autorise donc, sur le routeur A, les connexions sortantes vers un port 21. On doit de plus autoriser les connexions entrantes vers la machine *proxy*, à destination de *n'importe quel port supérieur à 1024*, et de numéro de port source égal à 20.

C'est cette deuxième règle de filtrage qui est dangereuse : il ne faut pas mettre de service TCP sur un port supérieur à 1024 sur la machine *proxy*, sous peine qu'il soit accessible par un éventuel pirate sur l'Internet. Or, il se trouve qu'il y a forcément un port supérieur à 1024 ouvert sur cette machine : il s'agit du port 3128 du *proxy*. Il faut donc configurer le logiciel *proxy* pour refuser les connexions autres que celles provenant du réseau privé, sur le port 3128, car les règles de filtrage des routeurs ne peuvent pas s'en charger sous peine de rendre le service FTP inaccessible, comme on vient de le constater.

Hébergement d'un serveur FTP

Pour héberger un serveur FTP, il faut le disposer sur le réseau bastion, et autoriser d'une part les connexions TCP depuis l'Internet vers le port 21 du serveur, et d'autre part les connexions TCP sortantes depuis le port 20 du serveur, vers un port quelconque d'une machine externe. Les connexions vers un port inconnu, problème inhérent au protocole FTP, ne sont donc ici pas un problème puisqu'elles sortent du réseau local.

Notons qu'il existe une fonctionnalité proposée sur certains serveurs, appelée *mode passive*, qui permet au client et non pas au serveur d'initier la connexion de transport des données. Si le client et le serveur FTP disposent tous les deux de cette possibilité, les problèmes de filtrage du service disparaissent. Mais de nombreux serveurs FTP sur l'Internet ne proposent toujours pas cette possibilité.

Une autre solution à ce problème est apportée par certains clients qui vont choisir le port de la deuxième connexion qui entre en jeu dans le transfert dans une plage bien déterminée, et plus réduite que la plage 1024-65535 en vigueur chez de nombreux clients, ou la plage 1024-4999 de certains autres clients du monde Unix. Habituellement, c'est la plage 40000-44999 qui est choisie, et il est alors facile de s'assurer qu'aucun service TCP n'écoute sur ces ports.

13.4.7 Services internes

Les services internes, par exemple un serveur WWW, sont disposés sur le réseau privé et ne nécessitent donc pas la mise en place de règle particulière.

13.5 TCP-Wrapper

13.5.1 Principe

Nous venons de décrire les moyens de filtrer les différents services non pas sur les machines qui les hébergent mais à travers les routeurs du réseau. Il arrive qu'on ne dispose pas de la possibilité de filtrage sur les routeurs du réseau, ou bien qu'on n'ait pas accès à la configuration de certains d'entre eux. C'est notamment le cas dans le cadre de l'accès par ligne spécialisée avec location du routeur et maintenance à distance par le fournisseur. En effet, lorsque le fournisseur loue le routeur de proximité et s'engage à maintenir l'accès à distance, il ne laisse habituellement pas l'accès de l'équipement à son client, afin d'éviter les fausses manipulations.

Dans ces différents cas de figure, lorsqu'on utilise Unix sur les serveurs du site, la solution du TCP-Wrapper est souvent très efficace. Cela consiste à filtrer les services TCP directement sur les machines qui les hébergent.

Pour cela, le service qu'on veut filtrer doit être activé par le démon `inetd`. Normalement, l'`inetd` écoute sur les ports des services standard qui lui sont indiqués par les fichiers de configuration `/etc/inetd.conf` et `/etc/services`. Lorsqu'une connexion intervient, l'`inetd` active le processus démon correspondant, par exemple `in.telnetd` dans le cadre d'une connexion de type `telnet`. On va donc intercaler un TCP-Wrapper entre les deux démons cités précédemment. Ce nouveau processus va consulter les fichiers de configuration `/etc/hosts.allow` et `/etc/hosts.deny` pour prendre la décision d'activer ou non le démon du service demandé. La figure 13.6 page ci-contre présente un exemple de connexion `telnet` avec un TCP-Wrapper.

13.5.2 Exemple de mise en place

Nous allons mettre en place un TCP-Wrapper pour interdire les accès `telnet` autres que ceux provenant de notre réseau local, sur notre serveur Unix.

On commence par rapatrier par FTP le logiciel TCP-Wrapper de Wietze VENEMA : il est disponible à l'URL :

```
ftp://ftp.ibp.fr/pub/unix/security/tcp_wrappers_7.2.tar.gz
```

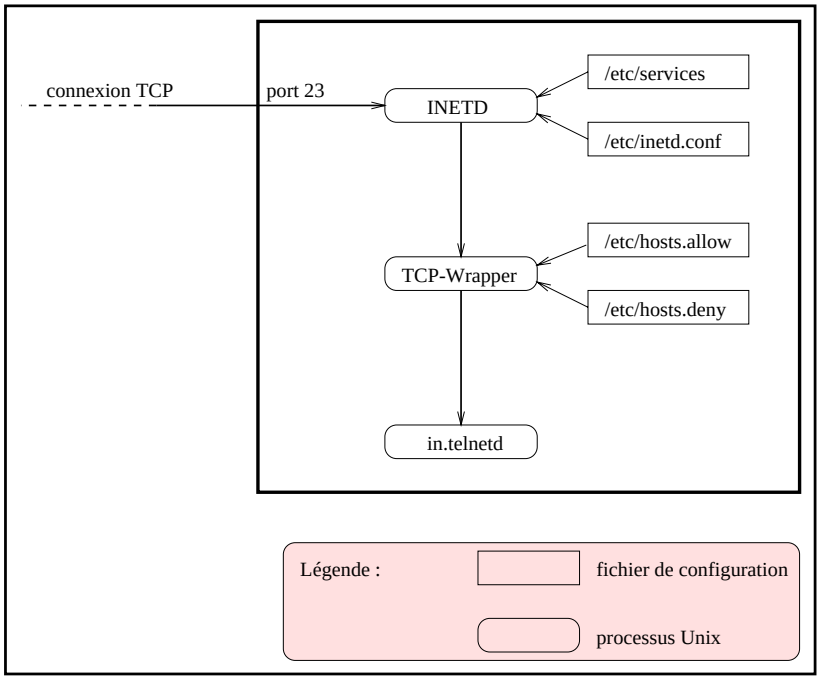

Figure 13.6 *TCP-Wrapper : exemple du service* `telnet`

```
<ls@fenetre> ftp ftp.ibp.fr
Connected to pascal.ibp.fr.
220 pascal FTP server (Version wu-2.4(4) Mon Feb 5 18:18:12 MET 1996) ready.
Name (ftp.ibp.fr:ls): ftp
331 Guest login ok, send your complete e-mail address as password.
Password:
230-
...
230 Guest login ok, access restrictions apply.
ftp> bin
200 Type set to I.
ftp> hash
Hash mark printing on (8192 bytes/hash mark).
ftp> cd /pub/unix/security
250 CWD command successful.
ftp> get tcp_wrappers_7.2.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for tcp_wrappers_7.2.tar.gz (94652 bytes).
############
226 Transfer complete.
local: tcp_wrappers_7.2.tar.gz remote: tcp_wrappers_7.2.tar.gz
94652 bytes received in 0.7 seconds (1.3e+02 Kbytes/s)
ftp> quit
221 Goodbye.
<ls@fenetre>
```

Avant de passer à la phase de compilation, il faut choisir le répertoire où le TCP-Wrapper va aller chercher les démons tel que `in.telnetd` pour `telnet`, ou `httpd` pour un serveur WWW.

Choisissons par exemple le répertoire `/usr/local/etc`.

Passons donc à la phase de compilation : il faut utiliser `make` avec pour premier paramètre

`REAL_DAEMON_DIR=/usr/local/etc`. Le deuxième paramètre est le type du système sur lequel on va installer le TCP-Wrapper. Il peut s'agir de : 386bsd, aix, alpha, apollo, convex-ultranet, dell-gcc, dgux, dgux543, dynix, epix, esix, freebsd, hpux, irix4, irix5, isc, linux, mips, ncrsvr4, netbsd, next, osf, ptx-2.x, ptx-generic, pyramid, sco, sco-nis, sco-od2, sunos4, sunos40, sunos5, sysv4, ultrix, unicos, unixware ou uxp.

Compilons donc sous Solaris :

```
<ls@fenetre> gzip -dc tcp_wrappers_7.2.tar.gz | tar xf -
<ls@fenetre> cd tcp_wrappers_7.2
<ls@fenetre> make REAL_DAEMON_DIR=/usr/local/etc sunos5
cc -O -DFACILITY=LOG_MAIL -DHOSTS_ACCESS -DPARANOID -DNETGROUP  -DGETPEERNAME_BUG \
 -DBROKEN_FGETS -DSOLARIS_24_GETHOSTBYNAME_BUG   -DDAEMON_UMASK=022 \
 -DREAL_DAEMON_DIR=\"/usr/local/etc\"    -DSEVERITY=LOG_INFO -DRFC931_TIMEOUT=10 \
 -DHOSTS_DENY=\"/etc/hosts.deny\" -DHOSTS_ALLOW=\"/etc/hosts.allow\"  -DTLI \
 -DALWAYS_HOSTNAME -c tcpd.c
cc -O -DFACILITY=LOG_MAIL -DHOSTS_ACCESS -DPARANOID -DNETGROUP  -DGETPEERNAME_BUG \
 -DBROKEN_FGETS -DSOLARIS_24_GETHOSTBYNAME_BUG   -DDAEMON_UMASK=022 \
 -DREAL_DAEMON_DIR=\"/usr/local/etc\"    -DSEVERITY=LOG_INFO -DRFC931_TIMEOUT=10\
 -DHOSTS_DENY=\"/etc/hosts.deny\" -DHOSTS_ALLOW=\"/etc/hosts.allow\"  -DTLI \
 -DALWAYS_HOSTNAME -c hosts_access.c
...
```

Installons maintenant, dans `/usr/local/etc`, l'exécutable créé : `tcpd`.

```
<ls@fenetre> su -
Password:
# mv tcpd /usr/local/etc/in.tcpd
# chmod 711 /usr/local/etc/in.tcpd
# chown root.daemon /usr/local/etc/in.tcpd
```

Il faut alors rechercher la ligne de configuration du service `telnet` dans le fichier de configuration `/etc/inetd.conf`. Il contient :

```
telnet  stream  tcp     nowait  root    /usr/sbin/in.telnetd     in.telnetd
```

On la remplace alors par :

```
telnet  stream  tcp     nowait  root    /usr/local/etc/in.tcpd   in.telnetd
```

Nous savons que `in.tcpd` va chercher les démons dans `/usr/local/etc`. Il faut donc qu'on recopie le démon `in.telnetd` dans ce répertoire :

```
# cp /usr/sbin/in.telnetd /usr/local/etc
```

Maintenant, il faut indiquer au démon `inetd` de relire le fichier de configuration qu'on vient de modifier. Il suffit pour cela de lui faire parvenir le signal `SIGHUP` :

```
# kill -HUP `/usr/bin/ps -ef | grep /usr/sbin/inetd | grep -v grep | awk '{print $2}'`
```

On doit, pour finir, créer les fichiers de configuration du TCP-Wrapper. Ces fichiers sont lus à chaque connexion. Ils sont destinés à décrire les filtres d'accès pour l'ensemble des services placés sous contrôle.

Nous voulons par exemple autoriser l'accès à `telnet` uniquement pour les machines dont l'adresse IP appartient au réseau de classe C `192.168.22.0` ou dont le nom appartient au domaine `fenetre.fr`.

Nous allons donc créer le fichier `/etc/hosts.allow` comme suit :

```
in.telnetd: ALL@.fenetre.fr ALL@192.168.22.
```

Pour indiquer qu'aucune autre machine n'a le droit de se connecter sur ce service, on construit le fichier `/etc/hosts.deny` comme suit :

```
in.telnetd: ALL@ALL
```

13.6 Autres outils

De nombreux autres outils pour la sécurité sont disponibles sur l'Internet. Certains sont fournis gratuitement avec leurs sources, qu'il suffit donc de recompiler sur ses machines, d'autres sont commercialisés.

On peut par exemple citer le logiciel `IPFW` permettant de transformer une machine BSD (FreeBSD, NetBSD, BSDi) en un puissant routeur filtrant, le logiciel `xinetd`[1] qui regroupe les fonctionnalités de l'`inetd` et d'un TCP-Wrapper et les nombreux outils TIS[2] (Trusted Information Systems).

Parmi les outils professionnels, on se doit de citer le produit Firewall-1 de CheckPoint Software Technologies[3], qui permet notamment de transformer une station Unix (SunOS, Solaris ou HP-UX) en routeur filtrant, et propose à l'utilisateur une interface conviviale pour définir la politique de filtrage le plus simplement possible.

Des fonctions de cryptographie sont également incluses dans ce produit, pour réunir en un réseau privé sécurisé des réseaux locaux interconnectés par l'Internet.

1. `ftp://ftp.ibp.fr/pub/unix/security`
2. `ftp://ftp.tis.com/pub/firewalls`
3. `http://www.checkpoint.com`

13.7 Vérifications

Il existe des outils qui permettent de simuler des tentatives de piratage provenant de l'Internet, afin de tester les filtres mis en place et le niveau de sécurité des serveurs présents sur le réseau. Les plus utilisés sont `ISS`[4] (Internet Security Scanner) et `SATAN`[5]. `ISS` est maintenant un produit commercial dont une version de démonstration est disponible sur l'Internet. `SATAN` est libre d'utilisation.

Ces outils possèdent un ensemble d'heuristiques pour tester les différents problèmes de sécurité bien connus dans le monde des réseaux TCP/IP.

Procédons à l'installation de `SATAN`. Notez que pour utiliser cet outil, il faut disposer d'une machine Unix, du langage Perl5[6] et d'un navigateur World Wide Web tel que Netscape Navigator.

On commence par rapatrier `SATAN` depuis `ftp.ibp.fr` :

```
<ls@fenetre> ftp ftp.ibp.fr
Connected to pascal.ibp.fr.
220-
220-    -- Bienvenue sur le serveur ftp de l'IBP et du CCR Jussieu --
220-Utilisez le compte 'anonymous' avec votre adresse e-mail comme mot de passe
220-    Merci de signaler les problemes eventuels a ftpmaint@ibp.fr.
220-
220-        -- Welcome on the IBP and CCR Jussieu ftp server --
220-    Please login as 'anonymous' with your e-mail address as password
220-             Please report problems to ftpmaint@ibp.fr.
220-
220-
220 pascal FTP server (Version wu-2.4(4) Mon Feb 5 18:18:12 MET 1996) ready.
Name (ftp.ibp.fr:ls): ftp
331 Guest login ok, send your complete e-mail address as password.
Password:
230-
...
230 Guest login ok, access restrictions apply.
ftp> bin
200 Type set to I.
ftp> hash
Hash mark printing on (8192 bytes/hash mark).
ftp> cd /pub/unix/security
250 CWD command successful.
ftp> get satan-1.1.1.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for satan-1.1.1.tar.gz (234999 bytes).
##############################
226 Transfer complete.
local: satan-1.1.1.tar.gz remote: satan-1.1.1.tar.gz
234999 bytes received in 1.6 seconds (1.4e+02 Kbytes/s)
ftp> quit
221 Goodbye.
```

Passons donc à la phase de compilation. Il faut commencer par lancer le script `reconfig`. Il

4. `ftp://ftp.ibp.fr/pub/cert/tools/iss`
5. `ftp://ftp.ibp.fr/pub/unix/security`
6. `ftp://ftp.ibp.fr/pub/perl/CPAN/src/5.0`

va se charger de détecter différents utilitaires sur le système et modifier les chemins d'accès correspondants dans les sources de la distribution :

```
<ls@fenetre> gzip -dc satan-1.1.1.tar.gz | tar xf -
<ls@fenetre> cd satan-1.1.1
<ls@fenetre> ./reconfig
checking to make sure all the target(s) are here...
Ok, trying to find perl5 now... hang on a bit...

Perl5 is in /usr/local/bin/perl5.001

changing the source in: bin/get_targets bin/faux_fping satan           \
 bin/boot.satan bin/dns.satan bin/finger.satan bin/ftp.satan           \
 bin/nfs-chk.satan bin/rex.satan bin/rpc.satan bin/rsh.satan           \
 bin/rusers.satan bin/showmount.satan bin/tcpscan.satan bin/tftp.satan \
 bin/udpscan.satan bin/xhost.satan bin/yp-chk.satan bin/ypbind.satan   \
 perl/html.pl

HTML/WWW Browser is /usr/local/bin/netscape

So far so good...
Looking for all the commands now...

Ok, now doing substitutions on the shell scripts...
Changing paths in config/paths.pl...
Changing paths in config/paths.sh...
<ls@fenetre>
```

À l'aide de la commande `make` suivie du nom du système d'exploitation sur lequel on va utiliser le logiciel, on va compiler la distribution. Les valeurs admises sont : aix, osf, bsd, bsdi, dgux, irix4, irix5, freebsd, hpux9, linux, sunos4, sunos5 et sysv4.

```
<ls@fenetre> make sunos5
cd src/misc; make "LIBS=-lsocket -lnsl" "XFLAGS=-DAUTH_GID_T=gid_t -DTIRPC" \
                                                    "RPCGEN=rpcgen"
cc -O -I. -DAUTH_GID_T=gid_t -DTIRPC  -c  md5.c
cc -O -I. -DAUTH_GID_T=gid_t -DTIRPC  -c  md5c.c
cc -O -I. -DAUTH_GID_T=gid_t -DTIRPC -o ../../bin/md5 md5.o md5c.o
cc -O -I. -DAUTH_GID_T=gid_t -DTIRPC -o ../../bin/sys_socket sys_socket.c
cc -O -I. -DAUTH_GID_T=gid_t -DTIRPC -o ../../bin/timeout timeout.c
cc -O -I. -DAUTH_GID_T=gid_t -DTIRPC -o ../../bin/rcmd rcmd.c -lsocket -lnsl
...
```

L'installation est terminée. On démarre donc SATAN, sous le compte root, par la commande suivante :

```
<ls@fenetre> su -
Password:
# setenv DISPLAY :0
# ./satan
SATAN is starting up....
```

SATAN lance alors Netscape Navigator et on voit s'afficher l'écran représenté sur la figure 13.7 page 423. Lorsqu'on veut tester les filtres, il faut lancer SATAN depuis un réseau en amont du firewall. Ce n'est pas utile lorsqu'on veut simplement tester la sécurité des démons présents sur les machines du réseau local. Pour notre exemple, nous allons supposer

que nous lançons SATAN sur la machine `fenetre.fenetre.fr` afin de tester le niveau de sécurité de la machine `cible.fenetre.fr` et les filtres des routeurs qui séparent ces deux équipements. Nous devons procéder pour cela aux étapes successives suivantes :

– On commence par sélectionner une base de données qui enregistrera les statistiques récoltées, comme présenté sur la figure 13.8 page 424. Cette base pourra être rechargée à un autre moment, afin d'être enrichie.
– On choisit alors une cible. Cela peut être un réseau entier ou une seule machine, comme l'indique la figure 13.9 page 425. On choisit, de plus, parmi trois niveaux d'intensité dans les tests.
– La figure 13.10 page 426 montre un exemple de rapport fourni par SATAN. Dans cet exemple, on constate que SATAN a découvert un ensemble de propriétés de la cible, ainsi que des problèmes de sécurité NFS. En cliquant sur les problèmes mentionnés, un écran d'explication ainsi que le moyen de les corriger apparaît.

Un des points forts de SATAN réside en la possibilité de repérer les relations de confiance entre les machines, pour pouvoir déterminer des chemins ouverts à travers le réseau, qui permettraient à un pirate de rentrer dans le réseau privé par bonds successifs.

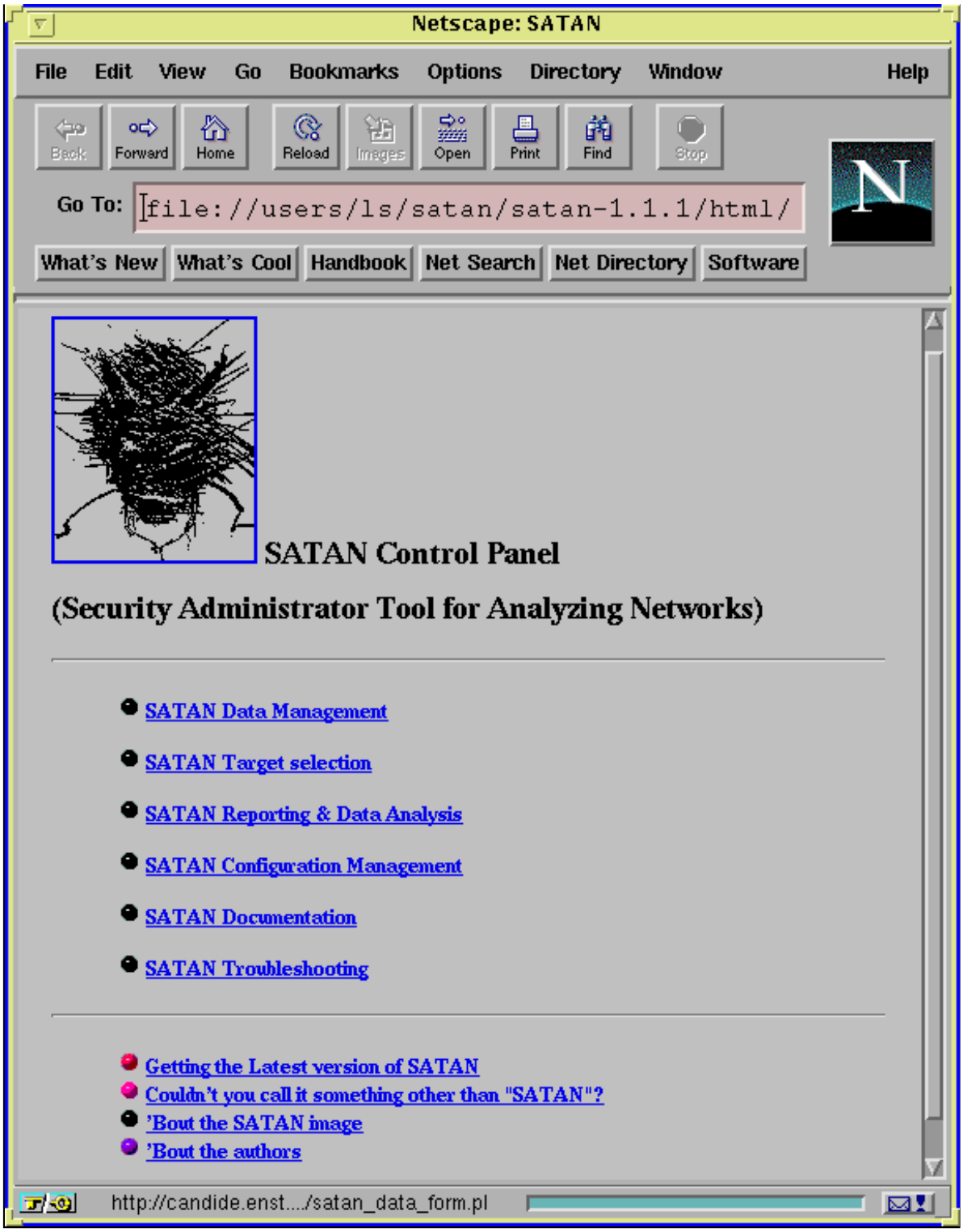

Figure 13.7 *SATAN : menu général*

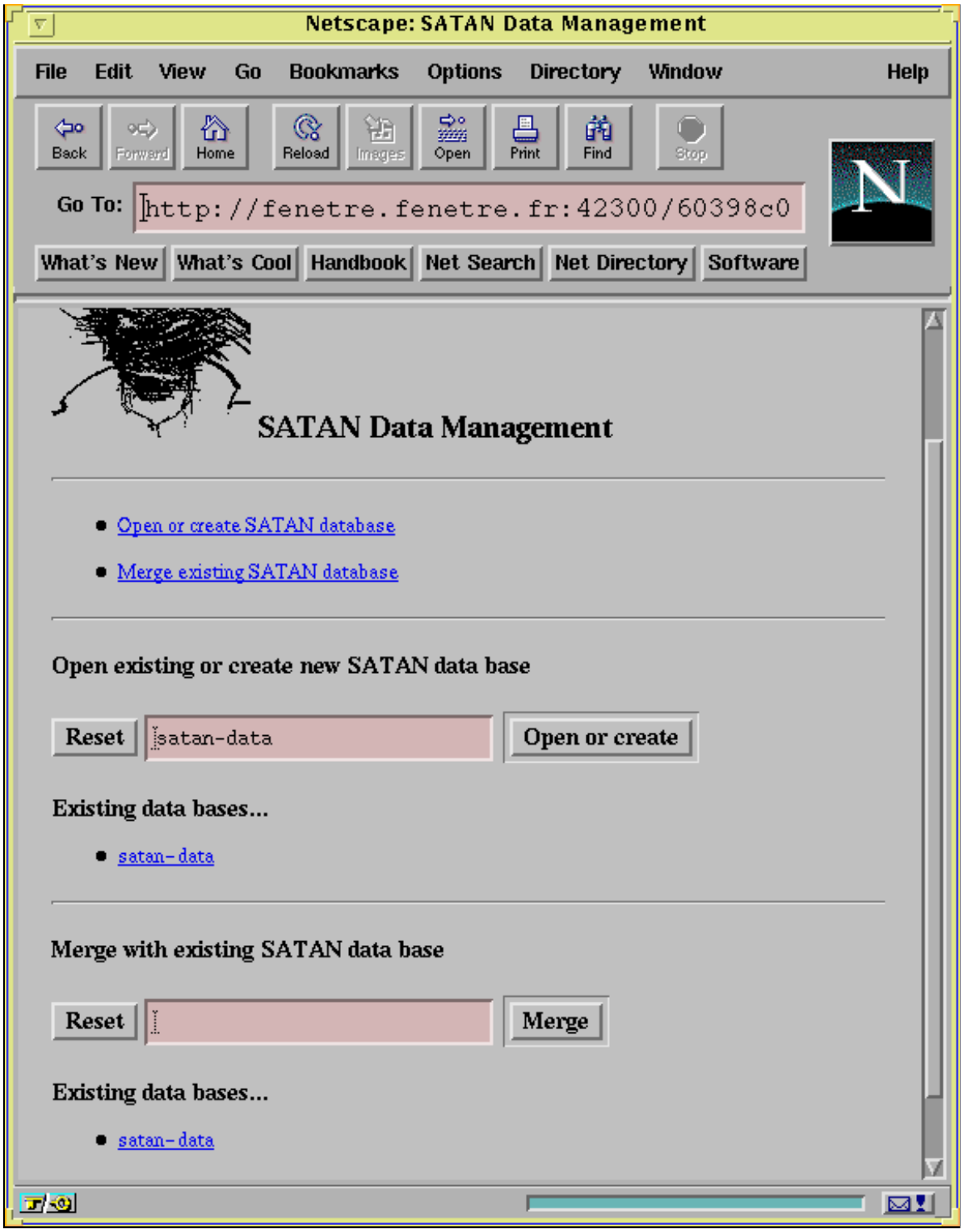

Figure 13.8 *SATAN: sélection d'une base de donnée*

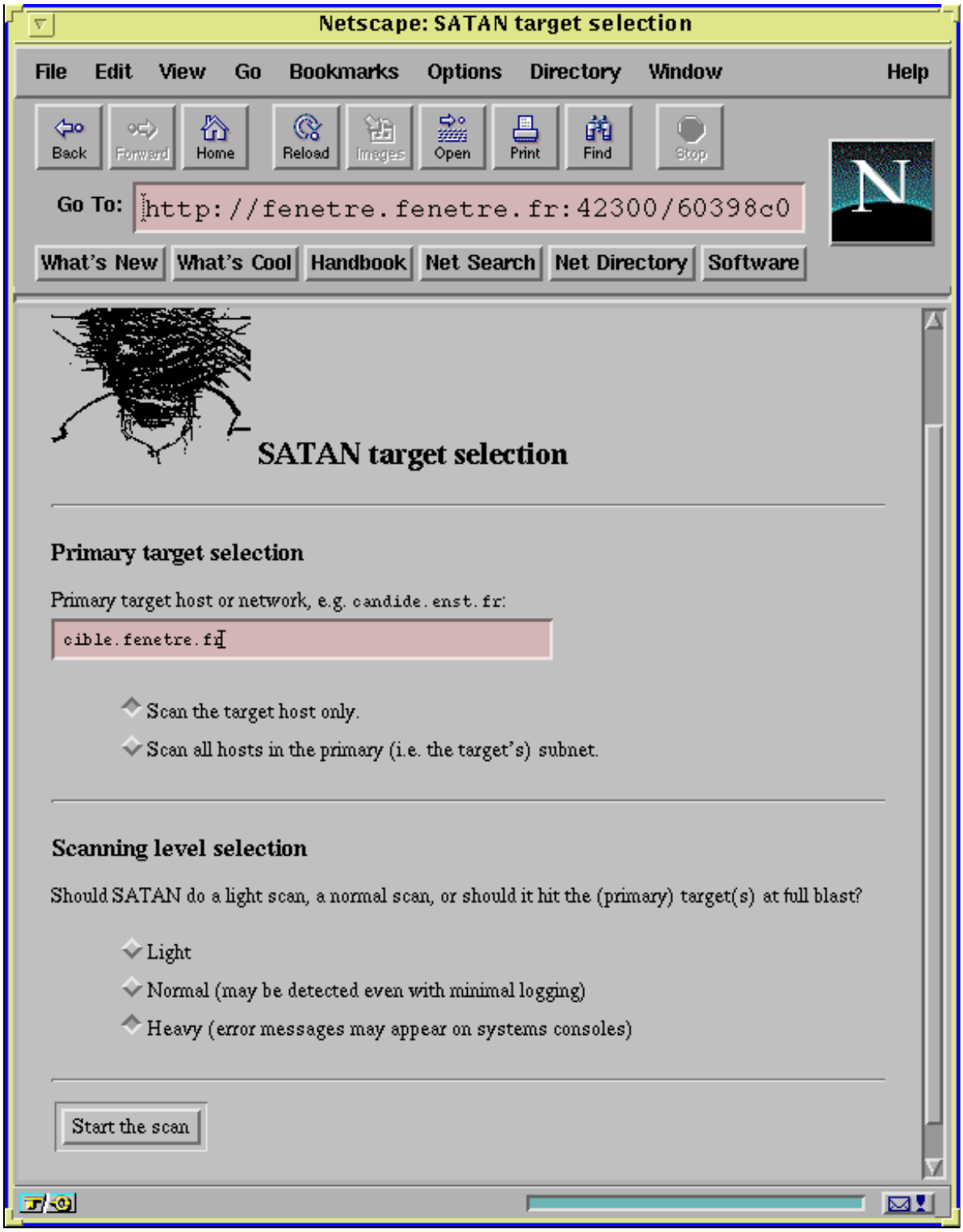

Figure 13.9 *SATAN : sélection d'une cible*

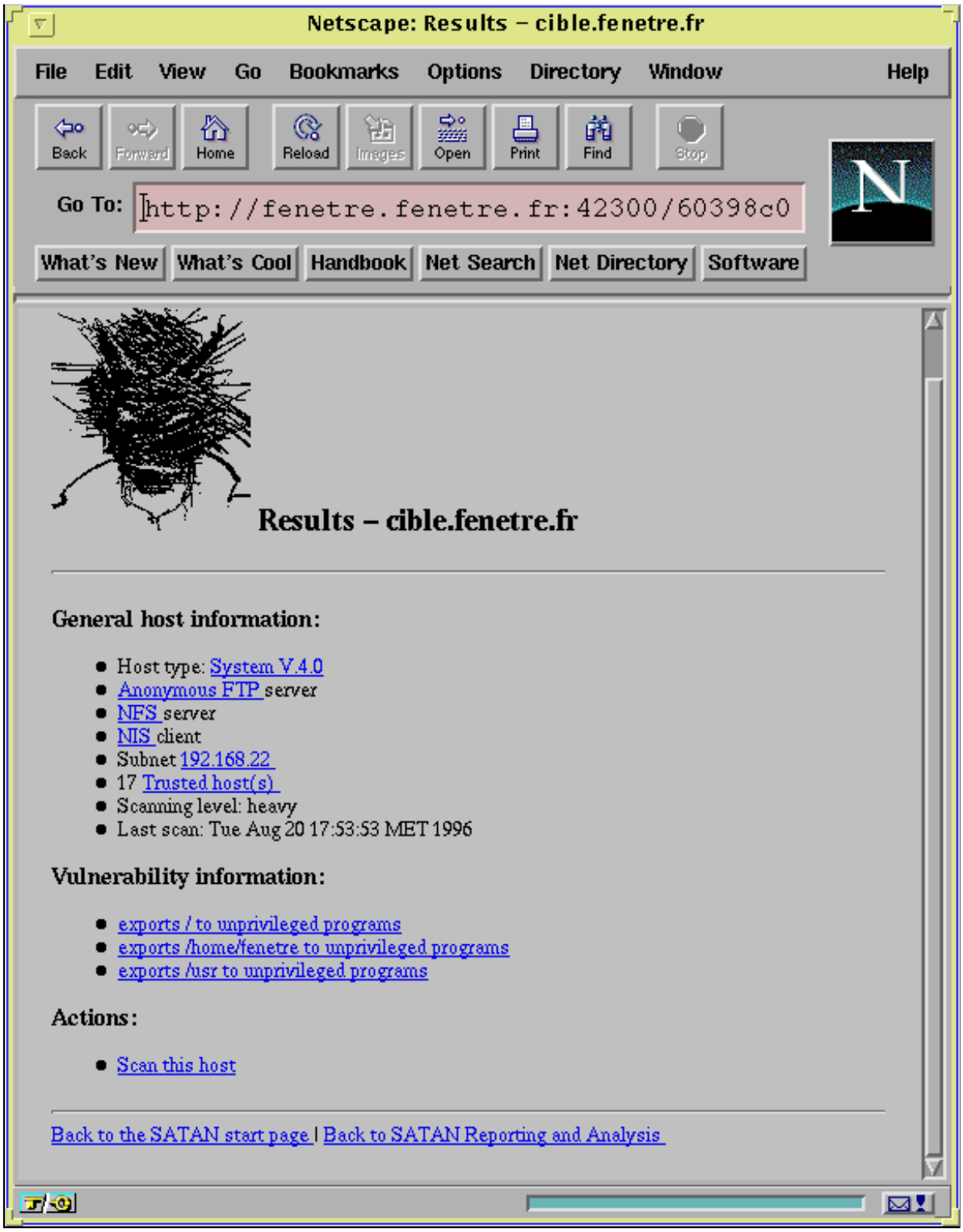

Figure 13.10 *SATAN : rapport du niveau de sécurité*

14

Les outils logiciels

L'émergence du commerce électronique nécessite des garanties de sécurité des données bien supérieures à ce que peut apporter l'Internet actuellement : il faut mettre les entreprises à l'abri de l'espionnage industriel ou des détournements de fonds, et protéger les clients d'éventuels vols et tromperies qui pourraient survenir à travers le réseau.

14.1 Notions de base

La sécurité sur l'Internet n'est pas un problème récent : comme le racontait déjà Cliff STOLL en 1987 dans *The Cuckoo's Egg*, les pirates sont apparus pratiquement dès les babultiements du réseau. La diversification de la population présente sur l'Internet et l'arrivée de services bancaires et commerciaux n'a pas augmenté le nombre d'actes de piratage sur l'Internet, mais leur teneur ; si les pirates agissaient auparavant par jeu, ils travaillent actuellement de plus en plus pour leur propre profit ou celui de tiers.

Comme tout système d'information, l'Internet permet malheureusement l'espionnage, la désinformation, le sabotage, le vol, l'imposture, etc. Les malveillances peuvent prendre des formes très variées, et les pirates informatiques ont toujours une longueur d'avance face à ceux qui cherchent à sécuriser leurs systèmes ou leurs réseaux, à moins que ces derniers ne se tiennent régulièrement à jour.

14.1.1 Les malveillances sur un réseau

Les **attaques passives** sont celles qui concernent la capture d'informations destinées à des tiers (écoute du réseau et accès à des données confidentielles) ainsi que l'analyse du trafic.

Les **attaques actives** sont les injections de fausses informations, la répétition de messages déjà envoyés, la modification de données destinées à d'autres et l'accès à un service, à une machine ou à une information sans autorisation.

Enfin, il y a **déni de service** lorsqu'une attaque rend impossible l'accès à une application. Pour cela, on peut saturer le réseau avec du trafic parasite, surcharger un serveur avec des requêtes à répétition ou encore détruire les tables de routage qui permettent aux paquets d'atteindre leur destination.

14.1.2 Sécurité des données dans un système informatique

Un système, connecté à un réseau ou non, doit pouvoir assurer une sécurité maximale des données, c'est-à-dire garantir :

- leur **confidentialité** : seules les personnes autorisées auront accès à l'information ;
- leur **intégrité** : les données n'auront pas été modifiées par une personne malveillante ou par accident.

Dans un système connecté à un réseau comme l'Internet, il faut de plus pouvoir authentifier les interlocuteurs, donc faire de l'**authentification** afin :

- de contrôler efficacement les accès au système (et donc limiter les risques de piratage) ;
- d'authentifier formellement l'auteur d'un message ou la provenance d'un fichier.

14.1.3 Pourquoi sécuriser un site ?

L'Internet est un réseau ouvert et relativement anarchique, dans lequel il faut se protéger par soi-même des intrus malveillants, à la différence de réseaux propriétaires comme Transpac ou les réseaux bancaires privés pour lesquels le fournisseur de service se doit d'assurer la sécurité des données.

La structure même de l'Internet, qui prévoit qu'un paquet peut emprunter n'importe quel chemin pour parvenir à sa destination, implique que n'importe quel nœud est un danger potentiel pour la confidentialité et l'intégrité dudit paquet. Un pirate, un espion ou une agence gouvernementale peuvent très bien intercepter les paquets qui transitent par leur site et les utiliser à mauvais escient.

14.2 Notions de cryptologie

Les techniques de cryptologie sont abordées à diverses reprises dans ce chapitre. Il est donc nécessaire d'en définir auparavant le vocabulaire et les bases techniques.

14.2.1 Quelques définitions

Le **cryptage** (ou chiffrement) est la procédure permettant de rendre inintelligible un message qui est alors qualifié de **crypté**. Le **décryptage** est l'opération inverse.

La **cryptographie** est l'art de transformer, à l'aide d'une convention tenue secrète, un message en clair en un message qui ne pourra être compris que par ceux qui connaissent la méthode permettant d'effectuer l'opération inverse.

La **cryptanalyse** est la recherche des différentes méthodes de décodage d'un système de chiffrement (et en particulier de ses faiblesses).

La **cryptologie** désigne la science qui regroupe l'étude de la cryptographie et la cryptanalyse.

14.2.2 Principes de base

Un système cryptographique peut se résumer par la formule suivante:

$$D\left(C\left(m\right)\right) = m \tag{14.1}$$

m étant le message en clair, C la fonction de cryptage et D la fonction de décryptage.

Une fonction de cryptage ou de décryptage est l'association d'un algorithme et d'une clé. L'algorithme est généralement rendu public afin d'être ouvert aux nombreuses analyses et critiques qui permettront de découvrir d'éventuelles failles tandis que la clé dépend de l'utilisateur de l'algorithme. La conservation du secret de la clé détermine la conservation du secret du message.

14.2.3 Efficacité d'un système cryptographique

De nombreux systèmes de cryptographie existent et sont disponibles librement. Certains sont basés sur une base scientifique sérieuse tandis que d'autres n'ont été testés que par un petit groupe et risquent d'être « cassés » relativement facilement par une ou plusieurs personnes suffisamment motivées. Il s'agit donc, avant d'utiliser le moindre système de cryptage, de s'assurer auprès de professionnels de ses fondements mathématiques.

Mathématiques et programmation

En premier lieu, l'algorithme mathématique doit être complexe, quelle que soit la méthode choisie (substitutions, transpositions, additions et multiplications avec modulos, transformations linéaires, permutations, etc.), le plus efficace étant de mélanger les différentes méthodes.

Comme la plupart des systèmes reposent sur des nombres choisis de façon aléatoire, il faut aussi pouvoir mettre en œuvre un générateur de nombres fortement aléatoires et non plus

de simples générateurs pseudo-aléatoires pour lesquels il devient facile de déterminer les nombres qui seront prochainement choisis. Les générateurs les plus efficaces se basent sur des phénomènes physiques (mesure du temps entre deux événements indépendants, comme la frappe de deux touches successives sur un clavier par exemple) et non plus de simples programmes.

Résistance aux attaques

Un système cryptographique est efficace s'il résiste entre autres aux tentatives de « craquage par la force[1] », c'est-à-dire aux méthodes de recherche systématique de toutes les combinaisons possibles dans l'espoir de tomber sur la bonne clé : il faut que le nombre d'essais soit prohibitif, du moins face aux moyens actuels (microprocesseurs rapides mis en parallèle), donc que les mathématiciens aient pu prouver que « l'algorithme est au moins équivalent à un problème dont on pense qu'il est difficile à résoudre » (au sens mathématique du terme).

Un tel système sera d'autant plus fiable qu'il saura faire face à ce type d'attaques même si un ou plusieurs messages cryptés et leurs contreparties en clair sont connus.

Enfin, un système devient très fiable si, en plus de tout cela, il résiste aux attaques différentielles, basées sur des statistiques effectuées sur des parties de messages cryptés.

14.2.4 Systèmes à clé privée

Principe des systèmes à clé privée

Un système à clé privée est un système totalement symétrique dans lequel la clé secrète sert au cryptage comme au décryptage :

$$D_k\left(C_k\left(m\right)\right) = m \tag{14.2}$$

m étant toujours le message en clair, C_k la fonction de cryptage, D_k la fonction de décryptage et k la clé privée.

Ce système implique que toutes les personnes susceptibles de recevoir ou d'émettre un message crypté doivent posséder la même clé privée. C'est la cryptographie dite traditionnelle.

Actuellement, les systèmes cryptographiques à clé privée sont les plus efficaces, les plus rapides et les plus facile à réaliser, d'un point de vue matériel comme d'un point de vue logiciel. Malheureusement, ils comportent deux inconvénients majeurs :

– la clé doit à tout prix rester secrète : il faut pouvoir la transmettre sur un canal sûr, ce qui est tout de même paradoxal et souvent difficile à réaliser ;

1. En anglais, *brute force attack*.

– l'authentification est très difficile dès que plus de deux personnes possèdent la clé privée : on peut prouver qu'on sait déchiffrer un message crypté et qu'on possède donc la clé, mais on ne peut pas s'identifier formellement.

Pour toutes ces raisons, les systèmes à clé privée sont très bien adaptés au cryptage interne ou personnel des fichiers (stockage sur un disque privé), mais pas à la transmission de messages confidentiels sur un réseau non sécurisé ni à l'authentification des correspondants.

Exemples d'algorithmes à clé privée

DES
L'algorithme DES (Data Encryption Standard), développé par IBM au début des années 70 à la demande du gouvernement américain, est une norme officielle depuis 1977.

Il opère sur des blocs de 64 bits avec des clés de 56 bits de long seulement, ce qui fait dire de lui qu'il n'est plus viable à l'heure actuelle en considération des puissances de calcul atteinte par les ordinateurs.

De plus, le bruit court depuis déjà une quinzaine d'années que la NSA (National Security Agency), agence américaine assurant la sécurité du pays) saurait « craquer » DES, ce qui a poussé des gens comme Philipp ZIMMERMANN à choisir IDEA plutôt que DES pour PGP (voir section 14.3.2 page 437).

D'autre part, DES n'est pratiquement jamais autorisé à l'exportation, alors que le gouvernement américain ne l'utilise même pas pour le cryptage d'informations de haut niveau de confidentialité.

IDEA
L'algorithme IDEA (International Data Encryption Algorithm) travaille aussi sur des blocs de 64 bits, mais avec une clé privée de 128 bits, ce qui garantit une meilleure résistance à la cryptanalyse. Il est basé sur le concept mathématique du mélange de différents groupes algébriques.

RC2 et RC4
Les deux algorithmes RC2 et RC4, développés par Ronald RIVEST, sont la propriété de la société américaine RSA Data Security Inc. Ce sont des fonctions à clé de taille variable, qui sont au moins aussi rapides que DES. Alors que RC2 fait du cryptage par blocs, RC4 travaille sur des flux de données. Un logiciel utilisant ces deux fonctions est exportable, à condition que la taille des clés soit limitée à 40 bits.

14.2.5 Systèmes à clé publique

En 1976, Whitfield DIFFIE et Martin HELLMAN annoncent qu'ils ont découvert une technique de cryptographie révolutionnaire, à clé publique, permettant d'échanger des messages cryptés sans avoir à rechercher un canal sûr pour transmettre une unique clé secrète.

Principe des systèmes à clé publique

Dans ce système, chacun possède une paire de clés, et non plus une seule clé à partager : la clé privée devra rester strictement secrète alors que la clé publique pourra être librement distribuée.

C'est un système asymétrique dans lequel le cryptage se fait à l'aide de la clé publique du destinataire, qui pourra seul le décrypter avec sa clé privée :

$$D_{K_s}\left(C_{K_p}(m)\right) = m \tag{14.3}$$

où m est le message en clair, C_K la fonction de cryptage, D_K la fonction de décryptage et K_p et K_s respectivement les clés publique et secrète du destinataire.

Signature électronique

Ce système offre un avantage important sur les systèmes à clé privée : l'authentification devient très facile. C'est le principe des signatures électroniques[2], rendu possible par le fait que les deux clés, privée et publique, sont totalement symétriques et interchangeables : l'expéditeur crypte un message avec sa clé privée et le destinataire pourra vérifier avec la clé publique correspondante que l'expéditeur est bien celui qu'il prétend être en décryptant le message. On aura donc une inversion de K_p et K_s par rapport à la formule 14.3 :

$$D_{K_p}\left(C_{K_s}(m)\right) = m \tag{14.4}$$

où, cette fois, K_p et K_s désignent respectivement la clé publique et la clé secrète de l'expéditeur.

Cependant, à chaque signature, on risque de découvrir un peu plus la composition de la clé privée de l'auteur du message, au risque de compromettre la sécurité des transactions futures ; il faut donc que le système soit si possible à apport de connaissance nulle[3] : même si on possède un grand nombre de messages signés d'un même expéditeur, il est pratiquement impossible de reconstituer sa clé privée à partir de fragments récupérés dans sa signature.

Un système à clé publique résout donc les problèmes posés par un système traditionnel, mais les systèmes asymétriques actuels sont beaucoup plus lents et sont inutiles dans le cas du cryptage de fichiers qui ne sont pas destinés à être transmis sur un canal non sûr.

Exemples d'algorithmes à clé publique

RSA L'algorithme RSA doit son nom à ses trois créateurs (Ronald RIVEST, Adi SHAMIR et Leonard ADLEMAN) qui ont depuis constitué ensemble la société RSA Data Security Inc. qui continue de développer de nouveaux algorithmes de cryptage.

2. En anglais, *digital signature*.
3. En anglais, *zero-knowledge system*

D'après ses trois auteurs, RSA est un algorithme équivalent au problème de la factorisation des grands nombres, réputé difficile. Il utilise l'exponentiation de grands nombres premiers pour obtenir une paire de clés fiable. Bien sûr, RSA n'est pas légalement exportable hors des États-Unis pour des clés de plus de 40 bits.

RSA et mathématiques

La fonction mathématique utilisée par RSA est relativement simple ; en reprenant les notations utilisées tout au long de ce chapitre et en ayant choisi un triplet (K_p, K_s, N) correspondant respectivement à la clé publique, la clé secrète et un nombre N connu des deux parties, on peut écrire que la fonction de cryptage $C_{K_p,N}$ est :

$$C_{K_p,N}(m) = \left(m^{K_p} \bmod N\right) \tag{14.5}$$

et la fonction de décryptage $D_{K_s,N}$ est :

$$D_{K_s,N}(m) = \left(m^{K_s} \bmod N\right) \tag{14.6}$$

Hors, il se trouve que K_p, K_s et N ont été choisis tels que K_p et K_s soient inversibles vis-à-vis de l'exponentiation à la puissance N, c'est-à-dire :

$$\forall x, \left(\left(x^{K_p} \bmod N\right)^{K_s} \bmod N\right) = x \tag{14.7}$$

On a donc :

$$\left(\left(m^{K_p} \bmod N\right)^{K_s} \bmod N\right) = m^{K_p K_s} \bmod N = m \tag{14.8}$$

ce qui correspond à (14.3) qui définit un système de cryptographie à clé publique.

El Gamal L'algorithme El Gamal, du nom de son inventeur, est quant à lui censé être équivalent au problème des logarithmes discrets, lui aussi considéré comme difficile.

14.2.6 Utilisation de systèmes de chiffrement à des fins d'authentification

Il est possible d'utiliser un système de chiffrement afin de vérifier l'identité d'un utilisateur qui se connecte sur un système. C'est notamment le cas des systèmes Unix.

Les systèmes Unix traditionnels enregistrent une version cryptée des mots de passe des utilisateurs dans un fichier appelé `/etc/passwd`. Ce fichier étant généralement lisible par n'importe quel utilisateur, il ne faut pas que ces informations chiffrées puissent être décodées par un utilisateur mal intentionné, et il faut que le système puisse les utiliser pour vérifier la validité d'un mot de passe entré au clavier.

Pour cela, au moment où l'utilisateur change son mot de passe, le système choisit deux sym-

boles au hasard, et utilise le mot de passe comme clé de cryptage pour chiffrer ces deux symboles. Le résultat est un mot chiffré de 13 caractères, dont les deux premiers sont les symboles tirés au hasard.

Voilà par exemple ce qui se passe lorsque l'utilisateur Luc STONED demande à changer son mot de passe :

1. l'ordinateur demande l'ancien mot de passe de Luc STONED et vérifie sa validité en utilisant la technique de vérification de mot de passe décrite plus bas ;
2. Luc STONED doit taper deux fois son nouveau mot de passe, pour être sûr qu'il ne commet pas d'erreur de frappe, ce qui l'empêcherait de se reconnecter. Dans notre exemple, l'utilisateur choisit le mot « denots » ;
3. l'ordinateur tire deux symboles au hasard : « ST » ;
4. il utilise la fonction `crypt` de la bibliothèque C du système pour chiffrer la chaîne « ST » avec, comme clé de chiffrement, « denots » ;
5. le résultat est « STcqxL1J8S2WQ ». Cette chaîne est insérée à l'intérieur du fichier `/etc/passwd` dans la ligne correspondant à l'utilisateur Luc STONED.

Maintenant, lorsque Luc STONED se connecte à nouveau sur cet ordinateur, voici ce qui se passe :

1. il tape son mot de passe « denots » au clavier ;
2. l'ordinateur extrait les deux premiers caractères de la chaîne « STcqxL1J8S2WQ » venant du fichier `/etc/passwd` : ce sont les deux symboles « ST » ;
3. il chiffre cette chaîne « ST » en utilisant le mot de passe « denots » tapé au clavier : cela lui donne « STcqxL1J8S2WQ » ;
4. cette chaîne étant identique à celle présente dans le fichier `/etc/passwd`, l'ordinateur considère que Luc STONED a tapé le bon mot de passe et le laisse se connecter.

Ce système permet dont d'identifier formellement l'utilisateur qui se connecte sans décoder le mot de passe crypté.

14.3 Techniques de sécurisation au niveau applicatif

14.3.1 Crack

Le programme `Crack` écrit par Alec MUFFETT a pour objectif de trouver le mot de passe Unix d'un utilisateur (voir page page précédente), en utilisant une méthode basée sur une recherche dans un dictionnaire. Ce programme est disponible gratuitement à l'UREC, à l'URL `http://www.urec.fr/Ftp/securite/Unix/Logiciels/` et son usage est très répandu parmi les pirates ; il faut donc qu'il soit régulièrement utilisé par les responsables informatiques qui souhaitent le rendre inefficace.

Principes de fonctionnement

`Crack` utilise pour ses recherches de mot de passe un ou plusieurs dictionnaires (au format « un mot par ligne ») contenant des mots susceptibles d'être utilisés par les utilisateurs du système.

Il tente ensuite, en utilisant une technique similaire à celle décrite page 433, de trouver le mot de passe en essayant successivement tous les mots du dictionnaire. Puis il essaye d'y rajouter des variations, comme de passer une lettre en majuscules[4] ou de retourner le mot.

Lorsqu'il essaye de trouver le mot de passe d'un utilisateur particulier, il ajoute temporairement dans son dictionnaire les informations qu'il peut trouver sur cet utilisateur, comme son nom et son prénom.

`Crack` dispose en standard d'un dictionnaire contenant les mots les plus couramment utilisés comme mot de passe, ainsi qu'un jeu de règles permettant de combiner ces mots au cours des différentes passes du programme ; dans ces règles, on trouve notamment le renversement d'un mot ou la concaténation de deux termes, ou encore le remplacement de la lettre « O » par un zéro.

Installation

Pour installer `Crack`, il faut tout d'abord décompresser et désarchiver le fichier récupéré, puis se rendre dans le répertoire nouvellement créé :

```
% zcat crack-4.1.tar.Z | tar xpBvf -
x Crack-4.1, 0 bytes, 0 tape blocks
x Crack-4.1/BUGS, 1112 bytes, 3 tape blocks
x Crack-4.1/DictSrc, 0 bytes, 0 tape blocks
x Crack-4.1/DictSrc/jargon, 3677 bytes, 8 tape blocks
[...]
x Crack-4.1/Crack, 6738 bytes, 14 tape blocks
x Crack-4.1/LICENCE, 4775 bytes, 10 tape blocks
% cd Crack-4.1
```

Ensuite, il faut copier le fichier de mot de passe qui nous intéresse, par exemple, sur un système traditionnel, `/etc/passwd` (ou seulement une partie de ce fichier), dans un fichier local, par exemple `monpass` :

```
% grep "Luc Stoned" /etc/passwd > monpass
% cat monpass
ls:STcqxL1J8S2WQ:512:1003:Luc Stoned:/u/ls:/usr/local/bin/zsh
```

Le script `Crack` doit être édité afin de définir la valeur de la variable `CRACK_HOME` qui doit pointer sur le répertoire courant.

4. Rappelons au passage que les mots de passe Unix sont sensibles à la différence entre les majuscules et les minuscules.

Utilisation

Le lancement en premier plan se fait avec l'option `-f` de `Crack`.

Le mieux est quand même de lire la documentation fournie avec `Crack` pour connaître toutes les options disponibles.

```
% ./Crack -f monpass
Invoked as: ./Crack -f monpass
Making dictionary Dicts/bigdict - This may take some time...
touch Dicts/.lockfile
Binary directory: /u/ls/Crack-4.1/generic
( cd ../Sources ; make clean )
make[1]: Entering directory '/u/ls/Crack-4.1/Sources'
[...compilation...]
Sorting data for Crack.
Flags: -f -i /tmp/pw.8133 Dicts/bigdict
Running program in foreground
pwc: Aug 27 15:32:04 Crack v4.1f: The Password Cracker, (c) Alec D.E. Muffett, 1992
pwc: Aug 27 15:32:04 Loading Data, host=serveur.fenetre.fr pid=8193
pwc: Aug 27 15:32:04 Loaded 1 password entries with 1 different salts: 100%
pwc: Aug 27 15:32:04 Loaded 240 rules from 'Scripts/dicts.rules'.
pwc: Aug 27 15:32:05 Loaded 74 rules from 'Scripts/gecos.rules'.
pwc: Aug 27 15:32:05 Starting pass 1 - password information
pwc: Aug 27 15:32:05 Guessed ls (/usr/local/bin/zsh in monpass) [denots] STcqxL1J8S2WQ
pwc: Aug 27 15:32:05 Closing feedback file.
pwc: Aug 27 15:32:05 FeedBack: 1 users done, 0 users left to crack.
pwc: Aug 27 15:32:05 FeedBack: information: all users are cracked after gecos pass
pwc: Aug 27 15:32:05 Done.
```

Comme on peut le voir, `Crack` a trouvé immédiatement le mot de passe de l'utilisateur Luc STONED. La ligne :

```
pwc: Aug 27 15:32:05 Guessed ls (/usr/local/bin/zsh in monpass) [denots] STcqxL1J8S2WQ
```

indique que le mot de passe est « denots ». `Crack` l'a trouvé car ce mot de passe est en fait le nom de l'utilisateur (« stoned ») à l'envers, ce qui correspond à une règle particulière se trouvant dans le fichier de règles livré avec le programme.

`Crack` dispose de plus d'une option `-m` permettant d'envoyer un courrier électronique à tous les utilisateurs dont le mot de passe a été trouvé. Cela permet de lancer `Crack` régulièrement (par une « crontab » par exemple) et d'avertir les utilisateurs automatiquement sans manipulation particulière.

Accélérer Crack

Il est possible d'installer, dans le répertoire de Crack, une bibliothèque disponible gratuitement appelée `ufc-crypt`, qui, sur certaines plates-formes, permet d'effectuer la recherche de mots de passe jusqu'à 30 fois plus rapidement qu'avec la fonction `crypt` du système. Sur certaines plates-formes (notamment Linux), la commande `crypt` disponible par défaut est déjà la plus rapide existante, donc l'utilisation d'une telle bibliothèque n'apporte rien de plus.

14.3.2 PGP

PGP (Pretty Good Privacy) est actuellement le programme de cryptage le plus répandu sur l'Internet. Il a été écrit en 1991 par Philipp ZIMMERMANN lorsque celui-ci s'est aperçu que la NSA cherchait à limiter l'usage des produits cryptographiques.

14.3.3 Principe de fonctionnement

PGP est un système de cryptographie hybride, en ce sens qu'il utilise à la fois les techniques de cryptographie à clé publique et à clé privée (voir page 430) afin de combiner la sûreté des premiers avec la rapidité des seconds. Quoi qu'il en soit, il apparaît à l'utilisateur comme étant uniquement un système de cryptographie à clé publique.

PGP est basé sur les deux algorithmes RSA (algorithme à clé publique, décrit page 432) et IDEA (algorithme à clé privée, cité page 431). La figure 14.1 schématise le processus utilisé : une clé IDEA I est tirée au hasard. Cette clé sera utilisée pour crypter le texte en clair en IDEA, et elle sera à son tour cryptée en RSA en utilisant la clé publique du destinataire et ajoutée au message chiffré.

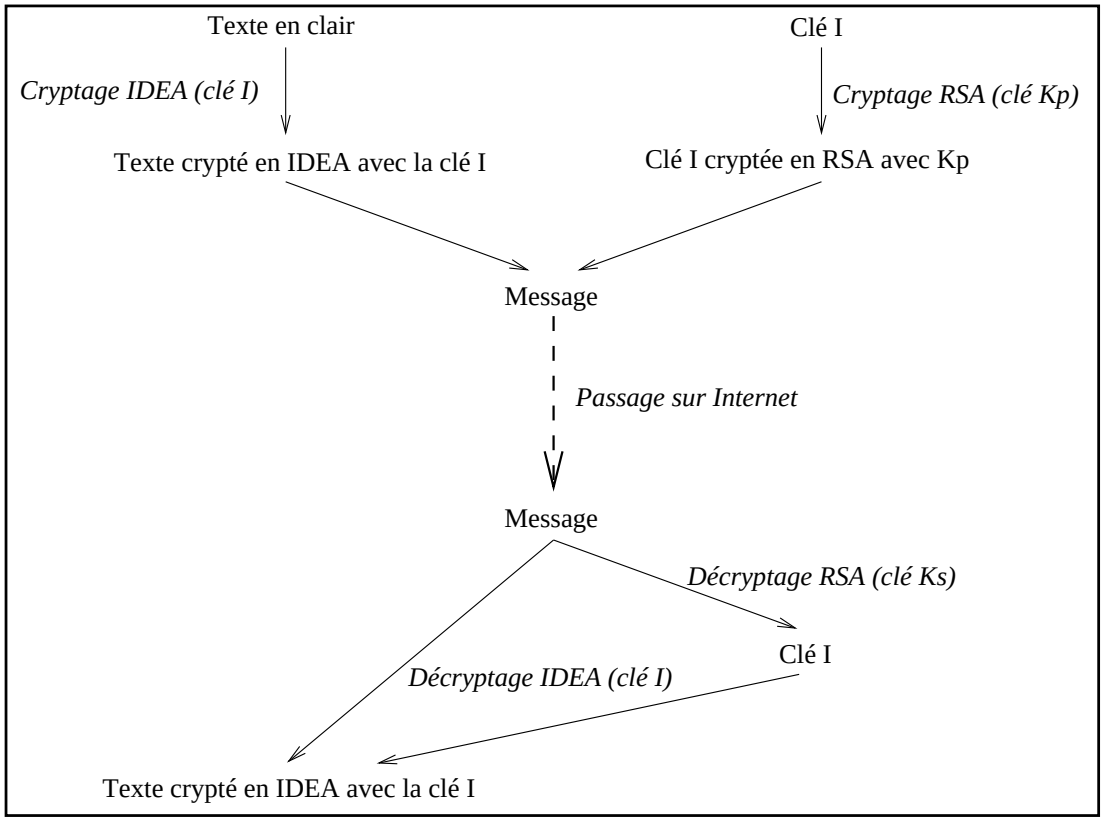

Figure 14.1 *Fonctionnement de PGP*

À la réception, le destinataire utilise sa clé privée pour décoder la clé I qui a servi à crypter le message, et déchiffre ce dernier avec la clé IDEA retrouvée.

14.3.4 Réseau de confiance

PGP fonctionne sur la notion de réseau de confiance[5] : cela signifie qu'un utilisateur A peut, s'il est sûr que la clé qu'il a sous les yeux est bien celle de l'utilisateur B, apposer sa signature sur la clé publique de B. Donc si Luc STONED connaît de manière sûre la clé publique de A et qu'il a confiance en lui, il acceptera la clé présentée comme étant celle de B, puisque signée par A.

De même, si Luc STONED signe la clé de A, A peut distribuer sa clé sur un canal non sûr : toute personne faisant confiance à Luc STONED et connaissant sa signature acceptera la clé de A comme étant authentique.

Signer sa propre clé

Il est possible, lorsque l'on utilise PGP, d'associer à la même clé plusieurs identités. Par exemple, si Luc STONED dispose d'une adresse électronique à son travail et une adresse électronique chez lui, il peut y associer la même clé PGP.
La clé publique étant, par définition, disponible publiquement, n'importe qui peut ajouter des identités à une clé existante et la redistribuer avec cette fausse identité. Pour cette raison, lorsque Luc STONED ajoute une nouvelle identité ou adresse électronique sur sa clé publique, il doit signer cette identité avec sa clé privée (il est le seul à pouvoir le faire puisque le seul à posséder cette clé privée).
Les dernières versions de PGP offrent la possibilité de signer automatiquement toute nouvelle identité générée lorsque l'utilisateur possède la clé privée correspondant à la clé publique concernée (option `AutoSign` du fichier de configuration de PGP).

14.3.5 Installation

PGP peut être obtenu, lorsque l'on se trouve hors des États-Unis et du Canada, à l'URL `http://www.ifi.uio.no/~staalesc/PGP/`. On y trouve les sources de PGP ainsi que des binaires pour de nombreuses architectures.

Nous n'allons pas détailler ici l'installation de PGP, car ce programme est tellement répandu qu'il est peu probable d'avoir à le compiler pour une plate-forme pour laquelle il n'a pas été testé. L'exécutable doit être copié dans un répertoire se trouvant dans le PATH des utilisateurs, et que chaque utilisateur désirant utiliser PGP devra créer chez lui un répertoire (par exemple `~/.pgp`) sur lequel pointera la variable d'environnement PGPPATH.

5. En anglais, *web of trust*

14.3.6 Configuration

La première étape pour utiliser PGP consiste en la création d'un fichier de configuration, appelé `config.txt`, qui doit se trouver dans le répertoire désigné par la variable d'environnement PGPPATH. Un exemple de ce fichier se trouve en général avec toutes les distributions, et il est conseillé de partir de cet exemple afin d'avoir les commentaires appropriés. Il faut également copier les deux volumes du guide utilisateur de PGP dans ce répertoire.

Le tableau 14.1 résume les différentes options modifiables et leur signification. La plupart des valeurs par défaut sont acceptables et peuvent ne pas être modifiées.

Option	Signification	Valeur conseillée
MyName	Nom par défaut de l'utilisateur	
Language	Langue utilisée par PGP	fr (français)
CharSet	Jeu de caractères	latin1
TMP	Répertoire temporaire	
Pager	Utilitaire de visualisation	less (si disponible)
Armor	Utilisation d'un codage ASCII	off
ArmorLines	Nombre maximum de lignes en ASCII	0
TextMode	Utilise le mode texte au besoin	on
ClearSig	Génère des signatures ASCII	off
KeepBinary	Conserve le fichier intermédiaire	off
Verbose	Niveau de trace	2
Compress	Compression	on
ShowPass	Affichage des mots de passe	off
Interactive	Mode interactif	on
EncryptToSelf	Crypter les messages pour le propriétaire	on
AutoSign	Signer les nouvelles identités	on
Legal_Kludge	Rester compatible	off
BakRing	Nom de l'anneau de secours	
Completes_Needed	Nombre de clés sûres nécessaires pour une nouvelle certification	1
Marginals_Needed	Nombre de clés moyennement sûres pour une nouvelle certification	2
Cert_Depth	Niveau de récursion de confiance	4
TZFix	Zone horaire	
Comment	Commentaire à ajouter dans les textes cryptés et/ou signés	
PubRing	Nom du fichier de clés publiques	
SecRing	Nom du fichier de clés privées	
RandSeed	Nom du fichier contenant la souche du générateur aléatoire	

Tableau 14.1 *Options de configuration de PGP*

L'étape suivante consiste en la génération d'une paire de clés. Cela se fait en utilisant l'option `-kg` de PGP et en se laissant guider par les instructions (en français lorsqu'on a choisi cette langue dans le fichier de configuration).

PGP demande à l'utilisateur d'entrer successivement :

1. un niveau de sécurité, en nombre de bits ; il est conseillé de choisir plus de 512 bits, 1024 étant la solution la plus lente mais la plus sûre ;
2. un nom d'utilisateur ;
3. un mot de passe, qui servira à protéger la clé secrète afin qu'un individu ayant accès au disque local ne puisse s'en emparer ;
4. un certain nombre de touches : les intervalles de temps les séparant serviront à construire la souche du générateur de nombres aléatoires.

```
marvin% pgp -kg
Pretty Good Privacy(tm) 2.6.3ia - Cryptographie à clé publique pour tous.
( c ) 1990-96 Philip Zimmermann, Phil's Pretty Good Software.  1996-03-04
Version internationale - ne pas utiliser aux Etats-Unis. N'utilise pas le
RSAREF.
Heure actuelle: 1996/11/30 13:33 GMT

Choisissez la taille de votre clef RSA:
    1)   512 bits- Niveau de base, rapide mais moins securitaire
    2)   768 bits- Niveau de securité eleve - vitesse moyenne
    3)  1024 bits- Pour les militaires, les diplomates... les paranoiaques
Choisissez 1, 2, ou 3, ou entrez le nombre de bits desires (Max 2048 bits): 3

Generation d'une clé RSA avec un module de 1024 bits.

Il vous faut un nom d'utilisateur pour votre clé publique. La forme
désirée pour ce nom d'utilisateur est votre nom, suivi de votre addresse
de courrier électronique entre <crochets>, si vous en avez une.
Par exemple:  Jean Dupont <dupont@toto.fr>
Entrez un nom d'utilisateur pour votre clé publique
(votre nom): Luc Stoned <luc.stoned@fenetre.fr>

Vous devez avoir un mot de passe pour protéger votre clé RSA
secrète. Votre mot de passe peut être n'importe quelle phrase ou portion
de phrase et peut avoir plusieurs mots, espaces, caractères de ponctuation
ou tout autre caractère imprimable.

Entrez votre mot de passe:
Entrez le même mot de passe de nouveau:
Notez que la génération de clé est une procédure lente.

Nous devons générer 939 bits aléatoires. Ceci est fait en mesurant
l'intervalle de temps entre les frappes de touches. Veuillez tapper du
texte aléatoire sur votre clavier jusqu'à ce que vous entendiez le
signal sonore:
   0 * -Assez, merci.
Le mot de passe est correct.  Un moment....
Certificat de signature de clé ajouté.
Génération de clé terminée.
```

Après cela, PGP a généré deux nouveaux fichiers dans le répertoire désigné par la variable d'environnement PGPPATH : `pubring.pgp` et `secring.pgp`. Il s'agit du fichier de clés publiques et du fichier de clés privées de l'utilisateur Luc STONED.

Il est ensuite possible de vérifier le contenu du fichier de clés publiques avec l'option `-kvv` de PGP :

```
% pgp -kvv
Pretty Good Privacy(tm) 2.6.3ia - Cryptographie à clé publique pour tous.
( c ) 1990-96 Philip Zimmermann, Phil's Pretty Good Software.  1996-03-04
Version internationale - ne pas utiliser aux Etats-Unis. N'utilise pas le
RSAREF.
Heure actuelle: 1996/08/30 13:42 GMT

Fichier de clé: '/u/ls/.pgp/pubring.pgp'
Type Bits/Clef     Date       ID utilisateur
pub  1024/71BA5CD9 1996/08/30 Luc Stoned <luc.stoned@fenetre.fr>
sig       71BA5CD9             Luc Stoned <luc.stoned@fenetre.fr>
1 clef trouvée.
```

14.3.7 Utilisation

Les commandes de PGP sont décrites dans le tableau 14.2.

Commande	Signification
`-kg`	Génération d'une nouvelle clé
`-ka`	Ajout d'une clé extérieure
`-kx`	Extraction d'une clé
`-kv`	Visualisation d'une clé
`-kvc`	Visualisation de l'« empreinte digitale »
`-kc`	Vérification en détail
`-kr`	Suppression d'une clé
`-ke`	Édition des paramètres de confiance d'une clé
`-ks`	Signature d'une clé
`-krs`	Suppression de la signature d'une clé
`-kd`	Révocation d'une clé
`-e`	Cryptage d'un message
`-s`	Signature d'un message
`-se`	Cryptage et signature d'un message
`-sb`	Génération d'une signature séparée
`-c`	Cryptage traditionnel (IDEA)
`-d`	Décryptage

Tableau 14.2 *Commandes de PGP*

Signature

Si Luc STONED désire signer le fichier `toto.txt` qu'il a créé, il utilisera la ligne de commande suivante :

```
% cat toto.txt
Je reconnais avoir détourné des fonds de la société FeNEtre Service
Express etc...

  Luc Stoned
% pgp -s -a toto.txt
Pretty Good Privacy(tm) 2.6.3ia - Cryptographie à clé publique pour tous.
( c ) 1990-96 Philip Zimmermann, Phil's Pretty Good Software.  1996-03-04
Version internationale - ne pas utiliser aux Etats-Unis. N'utilise pas le
RSAREF.
Heure actuelle: 1996/08/30 14:02 GMT

Une clé secrète est nécessaire pour faire une signature.
Vous devez avoir un mot de passe pour utiliser votre clé secrète RSA.
Clé pour le nom d'utilisateur: Luc Stoned <luc.stoned@fenetre.fr>
Clef de 1024 bits. Id clef 71BA5CD9 créé 1996/08/30

Entrez votre mot de passe: Le mot de passe est correct.  Un moment....
Fichier de signature en clair: toto.txt.asc
```

Voici le contenu du fichier `toto.txt.asc` généré par PGP :

```
-----BEGIN PGP SIGNED MESSAGE-----

Je reconnais avoir détourné des fonds de la société FeNEtre Service
Express etc...

  Luc Stoned

-----BEGIN PGP SIGNATURE-----
Version: 2.6.3ia
Charset: latin1

iQCVAwUBMib0X6E6npBxulzZAQHYCAQAmxnocF2K7zF9DXP69hWhuNGzW0kzMNcQ
pxgpBfPGXOWGIP5v0GIroQcGODiDvNpxGtT7O6vpfzqje8Kb0a4NIXSiSRoMnUE/
+Nb77iKj/NH/oyzjaAZdwra5/b+0WfC+Gg6/nDS2WgSaBOsoULxhl6nPXXOdugDj
5aW2Pk1tX9A=
=sQ7/
-----END PGP SIGNATURE-----
```

Le fichier `toto.txt.asc` contient la version signée du fichier `toto.txt`, c'est-à-dire que quelqu'un possédant la clé publique de Luc STONED peut dire, à coup sûr, que c'est bien Luc STONED qui a signé ce fichier.

La vérification se fait comme suit :

```
% pgp toto.txt.asc
Pretty Good Privacy(tm) 2.6.3ia - Cryptographie à clé publique pour tous.
( c ) 1990-96 Philip Zimmermann, Phil's Pretty Good Software.  1996-03-04
Version internationale - ne pas utiliser aux Etats-Unis. N'utilise pas le
RSAREF.
Heure actuelle: 1996/08/30 14:04 GMT
Ce fichier est signé. Une clef publique est nécessaire pour sa vérification.
.
Bonne signature de l'utilisateur "Luc Stoned <luc.stoned@fenetre.fr>".
Signature faite 1996/08/30 14:02 GMT en utilisant un clef de 1024 bits.
Id de la clef:71BA5CD9

Nom du fichier en clair: toto.txt
```

Cryptage

Le cryptage d'un fichier pour un destinataire ne nécessite pas la saisie d'un mot de passe (du moins tant qu'il n'est pas signé en même temps par l'expéditeur). Prenons l'exemple de Luc STONED qui désire envoyer un message au CERT (Computer Emergency Resource Team) :

```
% cat pb_securite
I have found the following security hole in my system:

I've noticed that the program called '/bin/su' is setuid root and then
insecure. However, when I remove the setuid bit, it doesn't work well.

For the time being, I cannot gain root access anymore and it's really
annoying. Do you have a solution with a non-suid su ?

Thanks in advance.

Luc Stoned, luc.stoned@fenetre.fr, System Administrator
```

Afin que le problème de sécurité découvert par Luc STONED ne transite pas en clair sur le réseau, celui-ci décide de le crypter à l'aide de PGP avec la clé publique du CERT :

```
% pgp -e pb_securite cert@cert.org
Pretty Good Privacy(tm) 2.6.3ia - Cryptographie à clé publique pour tous.
( c ) 1990-96 Philip Zimmermann, Phil's Pretty Good Software.  1996-03-04
Version internationale - ne pas utiliser aux Etats-Unis. N'utilise pas le
RSAREF.
Heure actuelle: 1996/09/10 17:15 GMT

La ou les clé(s) publique(s) du destinataire seront utilisées pour chiffrer.
Clé pour le nom d'utilisateur: CERT Coordination Center <cert@cert.org>
Clef de 1024 bits. Id clef 2DE30EC1 créé 1994/03/31
.
Fichier chiffré: pb_securite.pgp
```

Le fichier contenant le texte chiffré peut maintenant être envoyé par courrier électronique sans risque d'être déchiffré par un pirate qui ne connaîtrait pas la technique dont parle Luc STONED dans son message :

```
% Mail -s "Huge security hole" cert@cert.org < pb_securite.pgp
```

14.4 Techniques de sécurisation au niveau connexion

14.4.1 SSL

Le monde du WWW n'a pas encore adopté de norme de sécurité. C'est pourquoi quelques sociétés comme Netscape ont avancé des propositions, dont notamment SSL (Secure Socket Layer), qui présente l'énorme avantage d'être une norme publique au niveau connexion, c'est-à-dire qu'il permet de crypter n'importe quelle connexion en mode connecté, par exemple une

connexion établie à l'aide du protocole TCP, sans considération de l'application qui l'utilise. De plus, ce protocole permet d'authentifier de manière fiable le serveur et le client.

Étant donné que SSL fonctionne au niveau connexion, il permet le cryptage de sessions HTTP, telnet, ftp, etc. L'algorithme utilisé pour le cryptage est négocié entre le serveur et le client avant la transmission du premier octet de la communication réelle, ce qui permet d'assurer une totale confidentialité du dialogue.

Le canal SSL, une fois établi, assure trois propriétés :

1. Le canal est privé : le cryptage utilisé est un cryptage à clé privée, après avoir échangé cette clé de session à l'aide d'un cryptage à clé publique. Ceci est possible car la clé publique du serveur est toujours connue (voir deuxième point).

2. Le serveur est toujours authentifié, c'est-à-dire qu'on connaît sa clé publique. Il est possible de fonctionner en authentification faible, c'est-à-dire que le serveur la transmet au tout début de la connexion. Le client peut, selon la connexion, être ou non authentifié en utilisant le même procédé (cryptage à clé publique).

3. Le canal est sûr : des informations de redondance[6] sont utilisées afin de s'assurer de l'intégrité des données transmises.

Une fois que le canal crypté est établi, l'application peut l'utiliser de manière transparente ; cela permet notamment de n'avoir à modifier que quelques portions bien localisées d'applications pour pouvoir utiliser SSL. Le protocole SSL est utilisé dans le navigateur Netscape, le plus répandu actuellement sur l'Internet, ainsi que dans plusieurs serveurs WWW tels que le Netscape Commerce Server ou Apache SSL.

Lorsque Netscape utilise le cryptage SSL, le symbole représentant une clé jaune cassée en bas à gauche du navigateur (figure 14.2) est remplacée par le symbole représentant une clé entière sur fond jaune (figure 14.3).

Figure 14.2 *Clé cassée de Netscape*

Figure 14.3 *Clé entière de Netscape*

6. En anglais, *checksums*

14.4.2 SSH

Principe de fonctionnement

SSH est un programme permettant d'établir une connexion interactive sécurisée entre deux machines, tout en utilisant l'authentification pour s'assurer de l'identité de la machine distante. L'authentification se fait à la fois en utilisant les fichiers `.rhosts` et le cryptage à clé publique RSA. Comme dans le cas de SSL (voir page 443), aucune information ne circule jamais en clair. SSH a pour but de remplacer les trois programme `rsh`, `rlogin` et `rcp` par des équivalents sécurisés appelés respectivement `ssh`, `slogin` et `scp`, qui utilisent exactement la même syntaxe. Le serveur s'appelle `sshd` et écoute, par défaut, sur le port 22 (ce numéro de port peut être modifié dans le fichier de configuration du serveur).

De plus, les connexions X11 sont transmises sur le canal sécurisé et les authentifications X11 spécifiques sont transformées aux deux bouts afin qu'aucune information pouvant affaiblir la sécurité du serveur X11 ne soit apparente.

Des ports TCP arbitraires peuvent également être redirigés à travers SSH, dans les deux directions ; cela peut permettre notamment d'utiliser des protocoles de commerce électronique de manière sûre.

La clé utilisée par le serveur est régénérée toutes les heures (cette fréquence est configurable) et n'est jamais stockée sur disque, ce qui permet de contrer la plupart des actions extérieures de piratage. Chaque utilisateur peut de plus utiliser ses propres clés RSA afin de s'authentifier formellement.

Afin d'améliorer les temps de transfert, les données sont compressées avant de transiter sur le réseau, diminuant du même coup les probabilités de décodage des paquets par analyse statistique.

Installation

Le programme `ssh` se configure et s'installe automatiquement avec un ensemble de macro-commandes `autoconf`, c'est-à-dire qu'il n'y a aucun travail à faire pour la configuration et l'installation lorsqu'on souhaite accepter les valeurs par défaut proposées, qui sont tout à fait raisonnables :

```
% gunzip -c ssh-1.2.17.tar.gz | tar xpBf -
% cd ssh-1.2.17
% ./configure
[...]
% make install
```

Le programme `sshd` doit ensuite être relancé à chaque redémarrage de la machine : il faut pour cela modifier les fichiers d'initialisation de chaque ordinateur sur lequel doit tourner `sshd` afin de le prendre en compte.

Attention

Bien que `ssh` soit installé par défaut dans un répertoire traditionnellement partagé (`/usr/local`), il faut toutefois réexécuter la commande `make hostinstall` sur toutes les machines sur lesquelles on souhaite faire tourner le démon `sshd`, afin de générer une paire de clés différente pour chaque machine.

Les fichiers de configuration des serveurs peuvent être modifiés. Par défaut, `sshd` cherche son fichier dans `/etc/sshd_config`. Les options les plus couramment utilisées sont décrites dans le tableau 14.3.

Option	Signification
`Port`	Numéro de port sur lequel écoutera `sshd`
`HostKey`	Nom du fichier contenant la paire de clés de la machine
`ServerKeyBits`	Nombre de bits des clés du serveur
`KeyRegenerationInterval`	Temps de validité maximale d'une clé de session
`PermitRootLogin`	Permet de se connecter directement sous le compte système
`X11Forwarding`	Doit être activé afin de faire passer les connexions X11 sur la ligne sécurisée
`RhostsAuthentication`	Utilise les fichiers `.rhosts` des utilisateurs. Cela permet entre autres de rendre la migration de `rsh` vers `ssh` complètement transparente pour les utilisateurs
`PermitEmptyPasswords`	La désactivation de cette option protège le système contre les utilisateurs qui n'ont pas de mot de passe

Tableau 14.3 *Options de configuration de sshd*

Utilisation

L'utilisation de `ssh` et `slogin` apporte, en plus de la sécurité garantie par le cryptage de la connexion, d'autres avantages :

- la variable d'environnement `DISPLAY` est définie automatiquement sur la machine cible si elle l'était sur la machine source ;
- le programme peut demander un mot de passe en cas de besoin, contrairement à `rsh`.

La session ci-dessous montre comment l'utilisateur Luc STONED se trouvant sur la machine `station1.fenetre.fr` se connecte sur la machine `station2.fenetre.fr` en utilisant `slogin` :

```
% hostname
station1.fenetre.fr
% echo $DISPLAY
station1.fenetre.fr:0.0
% slogin station2
Password:
####################################
### Welcome on station2.fenetre.fr ###
####################################
% hostname
station2.fenetre.fr
% echo $DISPLAY
station2.fenetre.fr:1.0
```

Comme on peut le voir ici, le fait d'utiliser `slogin` plutôt que `rlogin` a modifié la variable d'environnement `DISPLAY` de Luc STONED qui pointe maintenant sur le port 6001 de la machine locale `station2.fenetre.fr` (le serveur X appelé « `:1.0` » correspond à ce numéro de port) sur lequel le programme `sshd` a mis en œuvre un relais qui redirige les connexions X sur le canal crypté.

14.5 Vérification de l'intégrité d'un système avec Tripwire

Les pirates informatiques utilisent des moyens de plus en plus sophistiqués pour se maintenir dans un système qu'ils ont su pénétrer ; il devient de plus en plus difficile de détecter leur présence ou leur passage. Il est donc nécessaire d'utiliser des outils qui vérifient l'intégrité des fichiers (notamment des exécutables) disponibles sur un site.

14.5.1 Principes de fonctionnement

Tripwire est un programme qui permet de s'assurer de la validité de fichiers et de répertoires en sauvegardant des informations de contrôle (« *checksums* ») redondantes dans un fichier qui servira ensuite à la vérification. Le schéma usuel d'utilisation de `Tripwire` est le suivant :

1. génération d'un fichier d'informations de contrôle pour les fichiers et répertoires sensibles ;
2. sauvegarde de ce fichier sur un support accessible uniquement en lecture, par exemple un disque dont on active la protection physique ou encore un CD-ROM ;
3. vérification ultérieure de ces informations de contrôle et comparaison avec la version sauvée précédemment ;
4. si ces informations diffèrent, il y a deux possibilités :
   (a) le propriétaire légitime du fichier ou du répertoire y a apporté des modifications : il faut mettre à jour le fichier contenant les informations de contrôle ;
   (b) le fichier ou le répertoire a été modifié par une source extérieure (acte de piratage ou accident) : il faut retrouver, à partir de bandes de sauvegarde par exemple, une version correcte du fichier ou du répertoire.

14.5.2 Installation

`Tripwire` peut être récupéré gratuitement sur le serveur WWW de l'UREC[7]. L'archive est signée avec `PGP` (voir la section sur PGP page 437). Après avoir décompressé et désarchivé le fichier, on peut en vérifier l'intégrité :

```
% zcat tripwire-1.2.tar.Z | tar xpvBf -
x Readme, 2967 bytes, 6 tape blocks
x T1.2.tar, 1048576 bytes, 2048 tape blocks
x T1.2.tar.asc, 282 bytes, 1 tape blocks
% pgp T1.2.tar
Pretty Good Privacy(tm) 2.6.3ia - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-03-04
International version - not for use in the USA. Does not use RSAREF.
Current time: 1996/08/28 15:40 GMT

File has signature.  Public key is required to check signature.

File 'T1.2.tar.$00' has signature, but with no text.
Text is assumed to be in file 'T1.2.tar'.
.
Good signature from user "Eugene H. Spafford <spaf@cs.purdue.edu>".
Signature made 1994/08/30 14:08 GMT using 1024-bit key, key ID FC0C02D5

Signature and text are separate.  No output file produced.
```

Comme PGP nous l'indique, le fichier a bien été signé par Eugene SPAFFORD, le 30 août 1994. Nous pouvons donc continuer l'installation en décompressant l'archive authentifiée :

```
% tar xpf T1.2.tar
% cd tripwire-1.2
```

Attention

Il est conseillé de lire attentivement le fichier `README` livré avec la distribution de `Tripwire`, car ce programme peut être configuré de manière différente selon la stratégie de protection qu'on souhaite adopter.

Nous allons maintenant détailler une installation standard de `Tripwire` sur système Solaris 2.5.1. Cela commence par une édition du fichier `Makefile` ; il faut s'assurer que les variables suivantes sont correctement définies :

`DESTDIR:` cette variable désigne l'endroit où les binaires seront installés. Idéalement, elle doit pointer vers un endroit qui sera accessible en lecture/écriture lors de l'installation et qui sera rendu physiquement inaccessible en écriture par la suite ;

`MANDIR:` l'endroit où seront placées les pages de manuel. `/usr/local/man` est un choix adopté traditionnellement ;

`CC:` le compilateur C. Si `gcc` est disponible, il est conseillé de l'utiliser plutôt que `cc`;

7. `http://www.urec.fr/Ftp/securite/Unix/Logiciels/Tripwire/`

CFLAGS: les paramètres supplémentaires passés au compilateur. `-O` sera utilisé pour activer l'optimisation ;

INSTALL: le programme d'installation. Sur Solaris, il faut choisir `/usr/ucb/install` ;

HOSTNAME: la manière d'obtenir des informations sur le nom du système. Pour Solaris, il faut choisir `uname -n`.

La deuxième phase consiste à repérer dans le répertoire `configs` la configuration la plus adaptée au système utilisé. Ici, pour Solaris, le choix le plus approprié est `conf-svr4.h`. Il faut reporter ce choix dans le fichier `include/config.h` :

```
/***
 *** Operating System specifics
 ***
 ***    Look in the .,/configs directory, and include appropriate header
 ***    file that corresponds with your operating system.
 ***/

#include "../configs/conf-svr4.h"
```

Dans ce même fichier, on peut configurer les chemins `CONFIG_PATH` et `DATABASE_PATH` qui désignent les répertoires contenant respectivement les fichiers de configuration du programme `Tripwire` et les bases constituées par les informations de contrôle. Idéalement toujours, ces variables devraient pointer vers des répertoires qui peuvent par la suite être configurés physiquement en lecture seulement.

De plus, on peut changer le nom des fichiers de configuration ou des bases de données ; l'inclusion d'un « `@` » dans un fichier de configuration fait que `Tripwire` substituera le nom de la machine sur laquelle il s'exécute à ce caractère. Par exemple, « `tw.db_@` » sera remplacé, si `Tripwire` tourne sur la machine `serveur.fenetre.fr`, par la chaîne « `tw.db_serveur.fenetre.fr` ».

Il suffit ensuite de lancer la compilation pour construire `Tripwire` :

```
% make
(cd aux; make CC=gcc CFLAGS="-O"  \
  LDFLAGS="" CPP="gcc -E" SHELL=/bin/sh all)
make[1]: Entering directory '/u/ls/tripwire-1.2/aux'
###
### Ignore warnings about shift count negative/too large on line 36
###
[...]
gcc -O -o tripwire config.parse.o main.o list.o ignorevec.o
   dbase.build.o utils.o preen.o preen.interp.o preen.report.o nullsig.o
   config.prim.o dbase.update.o config.pre.o help.o
   ../sigs/md5/md5wrapper.o ../sigs/md5/md5.o ../sigs/snefru/snefru.o
   ../sigs/crc32/crc32.o ../sigs/crc/crc.o ../sigs/md4/md4.o
   ../sigs/md4/md4wrapper.o ../sigs/md2/md2.o ../sigs/md2/md2wrapper.o
   ../sigs/sha/sha.o ../sigs/sha/shawrapper.o ../sigs/haval/haval.o
   ../sigs/haval/havalwrapper.o
gcc -O -o siggen siggen.c ../sigs/md5/md5wrapper.o ../sigs/md5/md5.o
   ../sigs/snefru/snefru.o ../sigs/crc32/crc32.o ../sigs/crc/crc.o
   ../sigs/md4/md4.o ../sigs/md4/md4wrapper.o ../sigs/md2/md2.o
   ../sigs/md2/md2wrapper.o ../sigs/sha/sha.o ../sigs/sha/shawrapper.o
   ../sigs/haval/haval.o ../sigs/haval/havalwrapper.o  nullsig.o utils.o
```

L'installation des exécutables et des pages de manuel se fait à l'aide de la commande `make install` ; il faut bien sûr que les répertoires de destination soient accessibles en écriture à ce moment précis.

14.5.3 Configuration et initialisation

Dans notre exemple, le fichier de configuration de `Tripwire` sera nommé `tw.config`. Pour commencer, on copie le fichier `lib/tw.config` livré avec `Tripwire` afin de l'utiliser comme point de départ :

```
% cp lib/tw.config tw.config
```

Le format du fichier de configuration est simple : chaque ligne est constituée d'un nom de fichier ou de répertoire qui devra être vérifié. Le caractère « `!` » placé avant ce nom signifie que ce fichier ou répertoire doit être ignoré par `Tripwire` tandis qu'un « `=` » permet de ne pas descendre récursivement dans un répertoire.

Ensuite se trouvent sur la ligne un ensemble de caractères indiquant les attributs à vérifier et à ignorer, précédés respectivement des caractères « `+` » et « `-` ». Le tableau 14.4 indique les significations des différents attributs.

Symbole	Signification
`p`	droits d'accès (permissions)
`i`	numéro d'inode
`n`	nombre de liens
`u`	identité du propriétaire
`g`	groupe du propriétaire
`s`	taille du fichier
`a`	date de dernier accès
`m`	date de dernière modification
`c`	date de création
`1`	première signature
`2`	seconde signature

Tableau 14.4 *Attributs de `Tripwire`*

Des attributs spéciaux ont été prédéfinis afin d'éviter de répéter inutilement les combinaisons les plus utilisées. Ils sont décrits dans le tableau 14.5 page ci-contre.

Un fichier pour vérifier l'intégrité du répertoire `/usr/local` et, récursivement, de son contenu à l'exception des fichiers de verrouillage créés par l'éditeur Emacs, pourrait être :

Symbole	Valeur	Signification
R	+pinugsm12-a	lecture seule
L	+pinug-sam12	fichier journal
N	+pinusgsamc12	tout vérifier
E	-pinusgsamc12	ne rien vérifier
>		ignore les fichiers de taille croissante

Tableau 14.5 *Combinaisons spéciales*

```
/usr/local      R
=/usr/local/lib/emacs/lock
```

La construction initiale de la base est faite à partir de la commande suivante :

```
% tripwire -initialize
### Warning:    creating ./databases directory!
###
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
### Phase 3:   Creating file information database
###
### Warning:   Database file placed in ./databases/tw.db_serveur.fenetre.fr.
###
###            Make sure to move this file file and the configuration
###            to secure media!
###
###            (Tripwire expects to find it in '/readonly'.)
```

Il s'agit maintenant de copier le fichier de configuration ainsi que la base de données nouvellement créée dans l'espace de sécurité, qui est rendu temporairement accessible en écriture :

```
% cp -p tw.config tw.db_serveur.fenetre.fr /readonly/
```

Il suffit maintenant de reprotéger physiquement le répertoire `/readonly` contre toute écriture, et le système `Tripwire` est prêt à fonctionner.

14.5.4 Utilisation quotidienne

Mis à part la phase d'initialisation, `Tripwire` fonctionne en trois modes différents :

vérification d'intégrité : en utilisant la base de données précédemment générée, `Tripwire` vérifie que les fichiers et répertoires présents sur le système n'ont pas été endommagés. Pour cela, il suffit d'invoquer `tripwire` sans paramètre ;

mise à jour de la base : dans ce mode, l'administrateur peut demander à `Tripwire` de remettre certains fichiers à jour dans sa base de données, d'en ajouter ou d'en ôter. Par exemple, pour ajouter le répertoire `/etc` dans la base de données, on peut utiliser la

commande `tripwire -update /etc` après avoir inséré la ligne correspondante dans le fichier de configuration de `Tripwire` ;

mise à jour interactive : dans ce mode, `Tripwire` demande à l'administrateur, pour chaque fichier modifié ou ajouté, s'il doit le mettre à jour dans la base de données. La commande pour choisir ce mode est `tripwire -interactive`.

Prenons l'exemple d'un répertoire `/usr/local` dans lequel `/usr/local/bin/zsh` a été mis à jour légitimement par l'administrateur système. Voila ce qui se passe lorsqu'on lance `Tripwire` en mode interactif :

```
% tripwire -interactive
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
### Phase 3:   Creating file information database
### Phase 4:   Searching for inconsistencies
###
###                     Total files scanned:         3017
###                           Files added:              0
###                           Files deleted:            0
###                           Files changed:            3
###
###                     After applying rules:
###                           Changes discarded:        2
###                           Changes remaining:        1
###
changed: -rwxr-xr-x root       498524 Aug 29 14:54:13 1996 /usr/local/bin/zsh
### Phase 5:   Generating observed/expected pairs for changed files
###
### Attr        Observed (what it is)         Expected (what it should be)
### =========== ============================= =============================
/usr/local/bin/zsh
       st_size: 498524                        393660
      st_mtime: Thu Aug 29 14:54:13 1996      Mon Dec 11 18:12:43 1995
      st_ctime: Thu Aug 29 14:54:13 1996      Mon Dec 11 18:12:43 1995
    md5 (sig1): 0d0JSJ9Ev9sqZ6G6Mw2BIh        2jCOZEDVtwyyPygKXtp43p
 snefru (sig2): 3qZQWWpXKWUl0vK5bkNgfZ        3x7pqF:I1z5UAsl9OpQNdd
---> File: '/usr/local/bin/zsh'
---> Update entry?  [YN(y)nh?] y

Updating entry: /usr/local/bin/zsh
### Updating database...
###
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
Updating: update file: /usr/local/bin/zsh
### Phase 3:   Updating file information database
###
### Old database file will be moved to 'tw.db_serveur.fenetre.fr.old'
###            in ./databases.
###
### Updated database will be stored in './databases/tw.db_serveur.fenetre.fr'
###            (Tripwire expects it to be moved to '/readonly'.)
###
###
### If you changed the tw.config file, remember to run 'twdb_check.pl' to
### ensure database consistency.
### See the README file for details.
```

14.6 État de la législation française

La législation française concernant l'utilisation des moyens de cryptographie a beaucoup évolué pendant l'été 1996. L'article 17 de la loi 96-659 du 26 juillet 1996[8] définit notamment la notion de « tiers de confiance[9] ».

14.6.1 La situation précédente

Jusqu'en juillet 1996, il était tout simplement interdit d'utiliser un outil pour chiffrer ses messages en France sans l'accord préalable du SCSSI, organe dépendant du Premier Ministre. Cette autorisation était systématiquement refusée dès lors que la demande concernait un logiciel permettant d'utiliser une technique « inviolable », comme par exemple PGP (voir page 437).

La signature électronique de messages était quant à elle soumise à simple déclaration ; par contre, les programmes utilisés pour signer des documents de manière sûre ne devaient pas permettre le cryptage, même si ces deux modes constituaient des fonctionnalités bien différenciées. Pour cette raison, PGP était également interdit d'utilisation pour la seule signature.

14.6.2 La situation actuelle

La loi 96-659 du 26 juillet 1996 a légalisé l'utilisation de moyens cryptographiques en y imposant une obligation de taille : tout utilisateur désireux de crypter ses données doit donner la clé permettant le décodage de ces informations à une « entité[10] » agréée par le Premier Ministre.

Cette notion de « tiers de confiance » est actuellement beaucoup critiquée par les entreprises qui travaillent avec des sociétés étrangères ; si l'entreprise française fait confiance à ce tiers, il n'en est pas obligatoirement de même pour ses partenaires étrangers qui peuvent craindre une interception de données commerciales sensibles par les services gouvernementaux français.

8. Cette loi a été publiée le 27 juillet 1996 au Journal Officiel.

9. En anglais *key escrow*.

10. Le terme entité utilisé ici est volontairement vague, les décrets d'application de cette loi n'ayant toujours pas été publié à l'heure de la mise sous presse de cet ouvrage.

CINQUIÈME PARTIE

Annexes

A

Fichiers de configuration de routeurs

La figure A.1 page suivante présente un exemple de fichier de configuration d'un routeur CISCO 1003 pour le raccordement par RNIS.

La figure A.2 page 459 présente un exemple de fichier de configuration d'un routeur CISCO 1005 pour le raccordement par ligne spécialisée.

La figure A.3 page 460 présente un exemple de fichier de configuration d'un routeur CISCO 1005 pour le raccordement par X25 sur Transpac.

```
version 11.0
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname routeurClient
!
username routeurFournisseur password leSecret
isdn switch-type vn3
!
boot system flash
enable password LucStone
!
!
interface Ethernet0
 description Interface sur le LAN
 ip address 192.168.22.33 255.255.255.224
 ip broadcast-address 192.168.22.63
 no ip route-cache
!
interface BRI0
 description Interface vers le fournisseur
 ip address 192.168.200.2 255.255.255.0
 ip broadcast-address 192.168.200.255
 no ip redirects
 encapsulation ppp
 dialer load-threshold 128
 ppp multilink
 ppp authentication chap
 dialer map ip 192.168.200.1 name routeurFournisseur 0123456789
 dialer-group 1
 dialer idle-timeout 30
!
dialer-list 1 protocol ip permit
!
router rip
 network 192.168.22.0
 passive-interface serial 0
!
ip default-network 192.168.100.0
ip route 192.168.100.0 255.255.255.0 192.168.200.1
ip route 192.168.200.0 255.255.255.0 192.168.200.1
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password EliBeurt
 login
!
end
```

Figure A.1 *Fichier de configuration d'un routeur CISCO 1003 pour RNIS*

```
version 11.0
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname Router
!
boot system flash
enable password LucStone
!
!
interface Ethernet0
 description Interface sur le LAN
 ip address 192.168.22.33 255.255.255.224
 ip broadcast-address 192.168.22.63
 no ip route-cache
!
interface Serial0
 description Interface sur ligne specialisee
 ip address 192.168.205.21 255.255.255.252
 ip broadcast-address 192.168.205.23
 no ip redirects
 bandwidth 64
!
router rip
 network 192.168.22.0
!
ip default-network 192.168.100.0
ip route 192.168.100.0 255.255.255.0 192.168.205.22
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password EliBeurt
 login
!
end
```

Figure A.2 *Fichier de configuration d'un routeur CISCO 1005 pour ligne spécialisée*

```
version 11.0
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname Router
!
boot system flash
enable password LucStone
!
!
interface Ethernet0
 description Interface sur le LAN
 ip address 192.168.22.33 255.255.255.224
 ip broadcast-address 192.168.22.63
 no ip route-cache
!
interface Serial0
 description Interface sur ligne specialisee
 ip address 192.168.210.2 255.255.255.0
 ip broadcast-address 192.168.210.255
 no ip redirects
 bandwidth 64
 encapsulation x25
 x25 address 175111111
 lapb T1 150
 x25 win 7
 x25 wout 7
 x25 ips 512
 x25 ops 512
 x25 nvc 2
 x25 idle 5
 x25 htc 16
 x25 map IP 192.168.210.1 175111101 ACCEPT-REVERSE
!
router rip
 network 192.168.22.0
 passive-interface serial 0
!
ip default-network 192.168.100.0
ip route 192.168.100.0 255.255.255.0 192.168.210.1
ip route 192.168.210.0 255.255.255.0 192.168.210.1
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password EliBeurt
 login
!
end
```

Figure A.3 *Fichier de configuration d'un routeur CISCO 1005 pour X25 sur Transpac*

B

Le formulaire du NIC-France

Le formulaire d'enregistrement ou de modification de domaine Internet dans la zone "`fr`" est à compléter et à faire parvenir par courrier postal au NIC-France, muni du cachet du prestataire de services.

Publié avec l'aimable autorisation du NIC-France.

NIC France

INRIA

Tel : +33 (1) 39 63 56 16
Fax : +33 (1) 39 63 55 34
E-Mail : nic@nic.fr
WWW : http://www.nic.fr/
Adresse : NIC France
Domaine de Voluceau
BP 105, F-78153 Le Chesnay CEDEX

Votre prestataire de service Internet
(doit être enregistré auprès du NIC France)

Formulaire d'enregistrement ou de modification de domaine Internet dans la zone 'fr'

Ce formulaire est valable jusqu'au 31 décembre 1996.
Ce formulaire contient deux pages, une annexe technique d'une page et une notice de deux pages.

1 - Organisme demandeur :

Nom complet : .....
Numéro : ..... Rue : .....
Ville : ..... Code Postal : ..... Pays : .....
Service : .....

Forme juridique : ❏ Société/organisme, numéro SIRET : _ _ _ _ _ _ _ _ _ _ _ _ _ _
❏ Association
❏ Autre : .....

2 - Nom de domaine :

Le NIC France reste seul juge quant à l'acceptation ou non du nom demandé conformément aux règles actuellement en vigueur dans l'Internet et à la **charte de nommage dans .FR**. Le NIC France n'a qu'un rôle d'enregistrement et n'est pas responsable des contestations sur l'utilisation d'un nom de domaine.

Pour définir votre nom de domaine, consultez le paragraphe 2 de la notice jointe en page 4.

Nom de domaine: _____ .FR

3 - Responsable administratif du domaine :

C'est la personne **au sein de votre organisme** qui est responsable du choix du nom de domaine. Elle autorise l'organisme à apparaître dans l'Internet sous le nom de domaine indiqué.

Nom : ..... Prénom : .....
Service et fonction : .....
Téléphone : +33 _ _ _ _ _ _ _ _ _ Fax : +33 _ _ _ _ _ _ _ _ _
Adresse de courrier électronique : .....

4 - Nature de la demande:

Cocher la case correspondant à la demande :

❏ **Création du domaine** (Celui-ci n'a jamais été enregistré auparavant) :
- ❏ La gestion du domaine sera assurée par l'organisme demandeur (remplir les rubriques 3, 6, Annexe).
- ❏ La gestion du domaine est confiée à un prestataire de service (remplir les rubriques 3, 5, 6).

❏ **Modification du** (des) responsable(s) administratif ou technique(s) (remplir les rubriques 3 et 6).

❏ **Changement de délégation de gestion :**
- ❏ Le domaine sera géré par l'organisme demandeur (remplir les rubriques 5, Annexe).
- ❏ La gestion du domaine est confiée à un prestataire de service (remplir les rubriques 5, Annexe).

5 - Délégation de gestion :

Si la gestion de votre domaine est confiée, ou a été confiée à un prestataire de service, indiquez ci dessous l'ancien et éventuellement le nouveau prestataire auquel vous confiez la gestion de votre domaine ou des serveurs de noms.
L'organisme demandeur s'engage à prévenir les prestataires concernés avant de demander toute modification au NIC France.

Ancien prestataire : .....
Nouveau prestataire : .....

6 - Contacts techniques :

Les contacts doivent être des personnes pouvant intervenir sur le fonctionnement des serveurs de noms. Si votre serveur de noms primaire est géré par un prestataire de service, un des contacts doit être une personne de l'équipe technique du prestataire de service.

Nom : ..... Prénom : .....
Société : .....
Téléphone : +33 _ _ _ _ _ _ _ _ _ Fax : +33 _ _ _ _ _ _ _ _ _
Adresse de courrier électronique : .....

Nom : ..... Prénom : .....
Société : .....
Téléphone : +33 _ _ _ _ _ _ _ _ _ Fax : +33 _ _ _ _ _ _ _ _ _
Adresse de courrier électronique : .....

7 - Engagements

- L'organisme s'engage sur l'exactitude des informations fournies dans ce document.
- Le demandeur a pris connaissance de la notice jointe à ce formulaire et s'engage à suivre les indications du NIC France quant à la gestion de son domaine, de ses sous domaines, et de ses serveurs de noms. Il s'engage à enregistrer dans sa zone les zones des autres services dépendant du même organisme qui lui en feraient la demande.
- L'organisme s'engage à notifier par écrit au NIC France tout changement dans la liste des serveurs, dans les personnes gérant le domaine de l'organisme, et généralement tout changement concernant des informations fournies dans ce document.
- Tout courrier électronique envoyé à la boite '`postmaster@votre-domaine.fr`' doit être remisable à une personne physique.

Fait à ....., Le .....

Signature du **responsable administratif** et cachet de l'organisme : .....
(**faire précéder de votre nom, de votre fonction dans l'organisme, et de la mention lu et approuvé**)

Signature du **prestataire** : .....

Ce formulaire doit être renvoyé par la poste à votre prestataire de service Internet.

NIC France

INRIA

Annexe technique au formulaire d'enregistrement de domaine Internet

Ce formulaire est valable jusqu'au 31 décembre 1996.
Si cette date est dépassée, vous pouvez obtenir une nouvelle version auprès du NIC France ou de votre prestataire de service Internet.

Cette annexe peut être remplie soit par l'organisme demandeur, soit par son prestataire de service.

Rappel du nom de domaine :

.FR

Serveurs de noms du domaine :

Indiquer en premier le serveur primaire, puis les serveurs secondaires. Un minimum de deux serveurs doit être indiqué (un primaire et un secondaire). Les noms doivent être complètement qualifiés (inclure votre nom de domaine).

L'organisme, ou son prestataire, s'engage à maintenir les serveurs de noms déclarés **fonctionnels et accessibles en permanence à partir de toutes les machines de l'Internet**.

Nom : ..... Adresse IP :

Nom : ..... Adresse IP :

Nom : ..... Adresse IP :

Nom : ..... Adresse IP :

Votre opérateur doit prendre contact avec le NIC France dès que vous êtes prêts à installer le domaine.

Nota :

- Les serveurs doivent être accessibles en permanence de toutes les machines de l'Internet. En particulier, tous les paquets UDP et TCP destinés au port 'domain' de numéro 53 ne doivent pas être filtrés. De plus, il est vivement conseillé que les paquets ICMP Echo ne soient pas filtrés.
- De manière à fournir un service fiable, les serveurs doivent de préférence ne pas se situer sur le même réseau physique, et il est conseillé que l'un des serveurs soit sur un accès différent des autres. Il vous est souvent possible de vous adresser à votre fournisseur de service ou à une organisation similaire à la vôtre pour lui demander d'être secondaire pour votre (vos) zone(s).
- Tous les serveurs sont interrogés de manière similaire. Les requêtes sont réparties entre les serveurs, sans distinguer le serveur primaire des serveurs secondaires.
- Il est indispensable de déclarer les zones inverses correspondant à vos réseaux. Cela permet d'obtenir le nom d'une machine à partir de son adresse IP. Cette déclaration doit être faite par le prestataire qui vous a fourni vos réseaux IP.

Standards de l'Internet :

Les RFC suivantes précisent le fonctionnement des serveurs DNS et des machines de l'Internet. L'organisme s'engage à respecter les standards établis dans l'Internet :

- RFC 1034, 1035 : Domain names - concepts & facilities, Domain names - implementation & specification.
- RFC 819, 821, 822, 920 : Domain Naming Convention for Internet User Applications, Simple Mail Transfer Protocol - SMTP, Standard for the format of ARPA Internet text messages, Domain Requirements, UUCP Mail Interchange Format Standard.
- RFC 1127 : A Perspective on Host Requirement for Internet Hosts.

NIC France

INRIA

Tel : +33 (1) 39 63 56 16
Fax : +33 (1) 39 63 55 34
E-Mail : nic@nic.fr
WWW : http://www.nic.fr/

Adresse : NIC France
Domaine de Voluceau
BP 105, F-78153 Le Chesnay CEDEX

Notice complétant le formulaire d'enregistrement ou de modification de domaine Internet dans la zone 'fr'

Cette notice est valable jusqu'au 31 décembre 1996.
Si cette date est dépassée, vous pouvez obtenir une nouvelle version auprès du NIC France
ou de votre prestataire de service Internet.

Le formulaire joint vous permet de faire la demande pour créer un domaine dans la zone 'fr', il est nécessaire au NIC France pour traiter votre demande. Ce formulaire doit être envoyé à votre prestataire par courrier postal, revêtu de la **signature du responsable administratif** et du **cachet de l'organisme**. Votre prestataire ajoutera sa signature et le fera suivre au NIC France.

Le NIC France ne traite que les demandes pour la zone '`fr`', il n'est pas habilité à effectuer des opérations dans les zones des autres pays ou dans les zones "organisationnelles", comme '`com`', '`net`', ou '`org`'...
Nous vous demandons de le remplir le plus lisiblement possible, et de compléter précisément toutes les rubriques. Votre prestataire peut vous aider à remplir les rubriques techniques. Dans les rubriques "informatiques", chaque lettre, chiffre, ou symbole a son importance !

Les informations de ce document font l'objet d'un traitement automatisé déclaré à la CNIL sous le numéro 350 376. La rectification des informations se fait auprès du NIC France.

1 - Organisme demandeur :

Il s'agit de décrire l'organisme (et non le prestataire) effectuant la demande. Cet organisme doit avoir une existence légale en France (société, administration, association, etc). Dans le cas où la demande est faite pour un service particulier de l'organisme, préciser le nom complet de ce service.

2 - Nom de domaine :

Le NIC France est seul juge quant à l'acceptation du nom de domaine demandé. Les noms sont attribués selon la règle **«premier arrivé, premier servi»**. Les contestations sur l'utilisation d'un nom de domaine sont résolues entre les parties concernées, le NIC France n'ayant qu'un rôle d'enregistrement.

Techniquement, le nom doit être composé à partir des caractères '`a`' à '`z`', du tiret '`-`', et des chiffres '`0`' à '`9`'. La longueur du nom choisi sera au minimum de 3 (trois) caractères. Aucune différence n'est faite entre les lettres majuscules et les minuscules.

2.1 - **Sociétés** (ou **organismes officiels**) : sont enregistrées directement sous '**`.FR`**'. Le nom de domaine alloué sera celui indiqué dans l'extrait KBIS du registre du commerce ou le document SIREN qui devra impérativement être fourni. Si l'organisme est connu sous le sigle usuel mentionné au KBIS, il pourra être utilisé comme nom de domaine.

2.2 - **Associations** : sont enregistrées sous le domaine '**`.ASSO.FR`**'. Joindre dans ce cas la copie de la parution au Journal Officiel ou le récépissé de déclaration à la Préfecture.

2.3 - Marques déposées : sont enregistrées sous le domaine '**.TM.FR**' (pour *trade mark*). Joindre dans ce cas le certificat d'enregistrement à l'INPI avec son numéro. "L'organisme déposant" enregistré à l'INPI doit être le demandeur du domaine.

2.4 - Publications (journal, magazine) : sont enregistrées dans le domaine '**.PRESSE.FR**'. Joindre dans ce cas la copie du document de la Bibliothèque Nationale portant le numéro ISSN.

2.5 - Services des Ministères : sont enregistrés dans le domaine '**.GOUV.FR**' après accord du SIG (Service d'Information du Gouvernement).

2.6 - Barreaux régionaux et cabinets d'avocats : sont enregistrés dans le domaine '**.BARREAU.FR**'

2.7 - Commissaires-priseurs : sont enregistrés dans le domaine '**.ENCHERES.FR**'. Joindre dans ce cas la lettre de la Chambre Nationale des commissaires-priseurs.

2.8 - Chambres de Commerce et de l'Industrie : sont enregistrées dans le domaine '**.CCI.FR**'. Joindre dans ce cas le certificat d'identification au Répertoire National des Entreprises et de leurs Etablissement (SIRENE).

Nota :
- Un seul nom de domaine par société/organisme est autorisé sous '.FR'.
- Dans le cas d'une demande d'enregistrement à l'INPI, le demandeur s'engage à faire connaître par écrit, sous 6 mois, la non acceptation de la marque. Dans ce cas le domaine sera supprimé.

3 - Responsable administratif de la connexion à l'Internet, ou du domaine :

C'est la personne au sein de votre organisme qui est responsable du choix du nom de domaine. Elle autorise l'organisme à apparaître dans l'Internet sous le nom de domaine indiqué. La présente demande, ainsi que les modifications ultérieures relatives à ce domaine doivent être signées par cette personne.

4 - Nature de la demande :

- **Création du domaine :**
  Il s'agit de créer un domaine qui n'a jamais été déclaré auparavant dans la zone '**fr**'.
- **Modification du domaine :**
  Pour modifier les contacts techniques et administratifs du domaine. Afin d'accélérer le traitement de la demande, le formulaire doit être signé par la personne ayant fait la demande précédente.
- **Changement de délégation de gestion :**
  Si la gestion technique du domaine est confiée à un prestataire de service, ou si vous changez de prestataire, il faut indiquer le nom du nouveau et éventuellement de l'ancien prestataire dans la rubrique "délégation de gestion". C'est le cas si parmi les serveurs de noms déclarés, un ou plusieurs d'entre eux sont des machines d'un prestataire de service.

5 - Délégation de gestion :

Vous devez remplir cette rubrique si vous confiez, ou retirez à un prestataire de services la gestion de votre domaine. Avant de transmettre la demande, **vous vous engagez à prévenir les prestataires concernés** des modifications demandées (résiliation par lettre recommandée avec accusé de réception).

6 - Contacts techniques :

Les personnes désignées (au minimum deux) doivent pouvoir intervenir sur le fonctionnement des serveurs de noms et les ressources associées à ce service (machines hôtes, routeurs, etc.). Si votre serveur de noms est géré par un prestataire de service, les contacts doivent être des membres de l'équipe technique du prestataire de service.

7 - Règles de comportement :

L'organisme doit utiliser des adresses de réseaux IP officiellement allouées par un "local registry" (un NIC ou un prestataire), RIPE, ou l'Internic, ou des adresses correspondant à la RFC 1918.
L'organisme doit éviter d'engorger les serveurs de noms avec des requêtes concernant des domaines inexistants (par ex : bitnet, uucp) en configurant correctement son serveur de noms et les logiciels utilisant celui-ci.
Le gérant du domaine a autorité pour créer des sous-entités et à en déléguer les droit à l'intérieur de son organisme, il reste responsable du bon fonctionnement de l'intégralité de son domaine. Ceci doit être fait en conformité avec les règles énoncées ci-dessus, après avis du gérant de 'fr' si cela s'avère nécessaire.
Un domaine appartenant à une entité administrative et juridique ne peut être utilisé par une autre entité administrative et juridique (**société-B.société-A.fr est interdit**).

C

Le formulaire de l'InterNIC

Le formulaire de l'InterNIC permet de réserver un domaine dans les zones "com", "net", "edu", "gov" et "org". Il doit être complété et envoyé par courrier électronique à l'adresse hostmaster@internic.net. On peut en récupérer un exemplaire à l'URL suivant : ftp://ftp.rs.internic.net/templates/domain-template.txt

```
[ URL ftp://rs.internic.net/templates/domain-template.txt ]         [ 8/96 ]

****************** Please DO NOT REMOVE Version Number ********************

Domain Version Number: 3.0

*************** Please see attached detailed instructions *****************

NOTE REGARDING ITEM 2 - SEE RFC1591 FOR DETAILS
    .COM is for commercial, for-profit organizations.
    .NET is for network infrastructure machines and organizations.
    .EDU is for 4-year, degree granting colleges/universities.
         (schools, libraries, museums register under country domains)
    .GOV is for United States federal government agencies.
         (state and local governments register under country domains)
    .ORG is for miscellaneous, usually non-profit, organizations.
         (orgs/individuals that do not clearly fit in any of the above)

Authorization
0a. (N)ew (M)odify (D)elete....:
0b. Auth Scheme................:
0c. Auth Info..................:

1.  Purpose/Description........:

2.  Complete Domain Name.......:

Organization Using Domain Name
```

```
3a. Organization Name.....:
3b. Street Address.....:
3c. City.....:
3d. State.....:
3e. Postal Code.....:
3f. Country Code.....:

Administrative Contact
4a. NIC Handle (if known).....:
4b. (I)ndividual (R)ole.....:
4c. Name.....:
4d. Organization Name.....:
4e. Street Address.....:
4f. City.....:
4g. State.....:
4h. Postal Code.....:
4i. Country Code.....:
4j. Phone Number.....:
4k. Fax Number.....:
4l. E-Mailbox.....:

Technical Contact
5a. NIC Handle (if known).....:
5b. (I)ndividual (R)ole.....:
5c. Name.....:
5d. Organization Name.....:
5e. Street Address.....:
5f. City.....:
5g. State.....:
5h. Postal Code.....:
5i. Country Code.....:
5j. Phone Number.....:
5k. Fax Number.....:
5l. E-Mailbox.....:

Billing Contact
6a. NIC Handle (if known).....:
6b. (I)ndividual (R)ole.....:
6c. Name.....:
6d. Organization Name.....:
6e. Street Address.....:
6f. City.....:
6g. State.....:
6h. Postal Code.....:
6i. Country Code.....:
6j. Phone Number.....:
6k. Fax Number.....:
6l. E-Mailbox.....:

Primary Name Server
7a. Primary Server Hostname....:
7b. Primary Server Netaddress..:

Secondary Name Server(s)
8a. Secondary Server Hostname..:
8b. Secondary Server Netaddress:

Invoice Delivery
9.  (E)mail (P)ostal.....:


An initial charge of $100.00 USD will be made to register the Domain name.
This charge covers any updates required during the first two (2) years.
```

The Billing Contact listed in Section 6 will be invoiced within ten (10) days of Domain name registration. For detailed information on billing, see:

ftp://rs.internic.net/billing/billing-procedures.txt
http://rs.internic.net/guardian/

The party requesting registration of this name certifies that, to her/his knowledge, the use of this name does not violate trademark or other statutes.

Registering a Domain name does not confer any legal rights to that name and any disputes between parties over the rights to use a particular name are to be settled between the contending parties using normal legal methods. See RFC 1591 available at:

ftp://rs.internic.net/policy/rfc1591.txt

By applying for the Domain name and through the use or continued use of the Domain name, the applicant agrees to be bound by the terms of NSI's then current Domain name policy (the 'Policy Statement') which is available at:

ftp://rs.internic.net/policy/internic.domain.policy

(If this application is made through an agent, such as an Internet Service Provider, that agent accepts the responsibility to notify the applicant of the conditions on the registration of the Domain name and to provide the applicant a copy of the current version of the Policy Statement, if so requested by the applicant.) The applicant acknowledges and agrees that NSI may change the terms and conditions of the Policy Statement from time to time as provided in the Policy Statement.

The applicant agrees that if the use of the Domain name is challenged by any third party, or if any dispute arises under this Registration Agreement, as amended, the applicant will abide by the procedures specified in the Policy Statement.

This Registration Agreement shall be governed in all respects by and construed in accordance with the laws of the United States of America and of the State of California, without respect to its conflict of law rules. This Registration Agreement is the complete and exclusive agreement of the applicant and NSI ("parties") regarding Domain names. It supersedes, and its terms govern, all prior proposals, agreements, or other communications between the parties. This Registration Agreement may only be amended as provided in the Policy Statement.

------------------------------ cut here ------------------------------------

GENERAL INSTRUCTIONS

Changes from version 2.0 of the Domain Template are:

- Security information has been added in Section 0 for use with MODIFY and DELETE submissions. For detailed information on Guardian, see:

  ftp://rs.internic.net/policy/internic/internic-gen-1.txt

- No changes can be made to Contact or Host records while using the MODIFY option on the Domain Template. For this purpose, Contact and Host Templates are available at:

  ftp://rs.internic.net/templates/contact-template.txt
  ftp://rs.internic.net/templates/host-template.txt

- The ability to register role Contacts in addition to individual Contacts has been added to Sections 4, 5, and 6.

- A separate field for a fax number has been added.

- A separate set of instructions has been added to facilitate the transfer of a Domain name from one organization to another.

Use the above Domain Template for registering new Domain names, for making changes to existing Domain name records, and for removing a Domain name from the InterNIC database and Root Servers. The template, and only the template, should be sent via email to HOSTMASTER@INTERNIC.NET. Please do not send hardcopy registrations to the InterNIC. Your Internet Service Provider (ISP) will be able to send email applications on your behalf if you are not connected. In the Subject of the message, use the words "NEW DOMAIN", "MODIFY DOMAIN", "DELETE DOMAIN", or "TRANSFER DOMAIN" appropriately, followed by the Domain name to help sort and locate incoming registration requests. In response to the submission of a template, you should receive an auto-reply with a tracking number. Use the tracking number in the Subject of any subsequent message you send regarding that registration action. When the registration is completed you will receive a notification via email.

Please DO NOT modify the template or remove the version number. The software that processes the template looks for an item number, followed by a period, followed by a colon. Information following the colon is compared with and inserted into the database. Please send only one template per message. When completing the template, make use of WHOIS at RS.INTERNIC.NET to check to see if the Domain name, organization name, Contacts, and Name Servers have been registered. Use that information where appropriate.

The instructions for completing each field are in the following four Sections - one each for NEW, MODIFY, DELETE, and TRANSFER. Please read the instructions carefully and make sure the template is properly completed to accomplish the action you desire.

REGISTERING A NEW DOMAIN NAME

Section 0 - Authorization

In item 0a, enter the character N or the word NEW to indicate a NEW Domain registration.

Items 0b and 0c are only REQUIRED for MODIFY and DELETE. These items will be ignored for NEW.

Section 1 - Purpose of Registration

Briefly describe the organization and/or the purpose for which this Domain name is being registered. The description should support the choice of top-level Domain in Section 2. If the Domain name is for an organization that already has a Domain name registered, describe the purpose of this Domain and why the additional name is needed. Indicate why existing names can not be used or why the proposed second-level name can not be used as a third-level name under a Domain name that is already registered.

Section 2 - Complete Domain Name

Top-level country Domains may be registered by inserting the two-letter country code in this Section. For a list of country codes, see:

ftp://rs.internic.net/netinfo/iso3166-countrycodes

See RFC 1591 for a description of the duties and responsibilities of top-level Domain administrators.

For second-level Domain names under COM, ORG, NET, EDU, or GOV, insert the two-part name of the Domain you wish to register. For example, EXAMPLE.COM. The total length of the two-part name may be up to 26 characters. The only characters allowed in a Domain name are alphabets, digits and "-". A Domain name CAN NOT begin or end with a "-" (see RFC 952). Consult RFC 1591 to determine the most appropriate top-level Domain to join. Briefly:

- COM is for commercial, for-profit organizations
- ORG is for miscellaneous, usually, non-profit organizations
- NET is for network infrastructure machines and organizations
- EDU is for 4-year, degree granting institutions
- GOV is for United States federal government agencies

US state and local government agencies, schools, libraries, museums, and individuals should register under the US Domain. See RFC 1480 for a complete description of the US Domain and registration procedures.

GOV registrations are limited to top-level US Federal Government agencies (see RFC 1816).

Section 3 - Organization using the Domain Name

The Domain name is considered to be registered to an organization, even if the organization is an individual. It is important in this Section to list the name and address of the end-user organization, not the provider organization.

If the organization has the same name as one that is already registered, explain this in Section 1 above.

Item 3b may be copied as necessary to reflect different lines of the street address. If item 3c, 3d, or 3e does not apply for your country, leave that item blank.

Item 3f MUST be the ISO two-letter country code. A list of ISO two-letter country codes is available at:

ftp://rs.internic.net/netinfo/iso3166-countrycodes

Section 4, 5, 6 - Contacts

The Administrative Contact is the person who can speak on behalf of the organization listed in Section 3. This person should be able to answer non-technical questions about the organization's plans for using the name and procedures for establishing sub-domains. See RFC 1032 for more detail on Administrative Contacts.

The Technical Contact is the person who maintains the Domain's Primary Name Server, resolver software, and database files. This person keeps the Name Server running and interacts with technical people in other Domains to solve problems that affect the Domain. The Internet Service Provider (ISP) often performs this role.

The Billing Contact will be sent invoices for Domain registrations and re-registrations. It is the Billing Contact's responsibility to assure timely payment.

If the Technical or Billing Contact is missing from the template, the Administrative Contact will be listed as filling those roles.

Each Contact in the InterNIC database is assigned a handle - a unique identifier to differentiate it from all other records in the database. Only one handle should exist for each individual or role.

If the handle is known, insert it in item "a" and leave the rest of the Section blank. If the handle is given and additional information is also provided, only the handle will be used. Any additional information will be ignored. Use handles whenever possible.

If the Contact handle is unknown or has never been registered, leave item "a" blank. The registration software will check for matches with existing Contact records. If a Contact record is found in the database that matches the information on the template in a significant way (name, and emailbox or phone), the database information will be assumed. Otherwise, the Contact will be registered. A Contact record CAN NOT be updated while using the Domain Template. To update Contact information, use the Contact Template available at:

ftp://rs.internic.net/templates/contact-template.txt

In item "b", indicate the type of Contact you are registering. If the Contact is an individual, enter I or INDIVIDUAL. If it is a group or organization where several individuals may be acting in that role, enter R or ROLE.

In item "c", enter the name of the individual or role being registered. If the Contact is an individual, the name should be entered as:

LASTNAME, FIRSTNAME MIDDLENAME

For example:

Smith, John X.

If the Contact is a role account, list the name of the role account in item "c" and the name of the organization the role account represents in item "d". Item "c" is REQUIRED for all Contacts. If the name of the role account is the same as that of the organization, fill in item "c" with that name and leave item "d" blank.

At least one line of address information is REQUIRED in item "e". This address should be the postal mailing address for any correspondence directed to the Contact. Enter the city and the standard two-letter state abbreviation in items "f" and "g" respectively if the Contact is located in the United States.

You may add as many lines as necessary for your street address. You may do so by repeating item "e".

For the U.S. addresses, the postal code is REQUIRED in item "h".

Enter the ISO two-letter country code in Item "i". If no country code is entered, the Contact is assumed to be located in the United States. A list of ISO two-letter country codes is available at:

```
      ftp://rs.internic.net/netinfo/iso3166-countrycodes

   Every Contact is REQUIRED to provide at least one valid phone number in
   item "j".  For informational purposes, you may supply a fax number in
   item "k".

   A valid Internet email address is REQUIRED for all Contacts in item "l".
   Separate the username and hostname parts of the mailbox by an "@" symbol.

   For example:

      user@example.com


Section 7, 8 - Name Servers

   At least two independent Servers MUST be provided for translating names
   to addresses for Hosts in the Domain.  A Domain name may be removed from
   the InterNIC database, after notice, if at least two Name Servers are
   not reachable and functioning properly.

   DO NOT list Name Servers if you do not have permission from the owners
   to do so.  Listing Name Servers without the explicit approval of the
   owners is not only unethical, but can cause operational problems for
   the Name Servers listed.

   The Servers should be in physically separate locations and on different
   networks, if possible.  The Servers should be active and respond to
   Domain Name System (DNS) queries BEFORE this application is submitted.
   Incomplete information in Sections 7 and 8 will result in a returned
   template.  Most ISPs can provide one or more Name Servers if you do not
   have your own.

   Please provide the Fully Qualified Domain Name (FQDN) of the Host that
   is to be the Name Server.  For example, enter NS.EXAMPLE.COM and not
   just NS.

   If several Secondary Servers are desired, copy Section 8 as many times
   as needed.  Please DO NOT renumber or change the copied Section.

   The registration software makes a cross check between the listed pairs
   of Host names and IP addresses to see if there are any matches with
   either in the database.  If a match with an IP address in the database
   is found, the name in the database will be assumed.  If a match with a
   Host name in the database is found, the IP address will be assumed.
   If neither match a Host record in the database, the Name Server on the
   template will be registered, assigned a handle, and entered into the
   database.

   Neither the name nor IP address of a registered Name Server will be
   changed as a result of a Domain registration.  A registered Name
   Server's name or IP address can only be changed by submitting a
   Host Template available at:

      ftp://rs.internic.net/templates/host-template.txt


Section 9 - Invoice Delivery

   If you wish to receive your invoice electronically, enter the character
   E or the word EMAIL in item 9.  If you wish to receive your invoice by
   postal mail, enter the character P or the word POSTAL in item 9.
```

MODIFYING A DOMAIN NAME RECORD

Changing an existing record is done by replacement. That is, the contents of various fields in the database are replaced with new information from the template. If the modification involves first registering a Contact or Name Server that is not in the database, the instructions for completing Sections 4-8 in "REGISTERING A NEW DOMAIN NAME" apply. Use WHOIS if you are not sure about the current information for a Domain, Name Server, or Contact.

The Domain name itself CAN NOT be modified. To change the Domain name, send two templates - the first to register the new name and the second to remove the old name when it is no longer needed.

Changes will/will not be made according to the security parameters established by the Contacts for the Domain and the entries in items 0b and 0c. If the Contacts have not chosen any level of security, the change will be made if the template comes from a reasonable source. This source may be from a listed Contact for the Domain, others in the same organization, the current provider, or a new provider that is about to provide support for the Domain.

Likewise, notification of pending or finished changes will be made according to the security parameters chosen by the Contacts. If no security has been chosen, the notification of the change and the approximate time the change will take effect will be sent to:

- the requester
- if Contacts are changing, both old and new Contacts
- if Name Servers are changing, the Technical Contacts for the Domains in which the old and new Primary Name Servers reside

This notification will help ensure that all parties involved are aware of, and agree with, the change.

Section 0 - Authorization

In item 0a, enter the character M or the word MODIFY to indicate a modification to an existing Domain registration. Transfer of a name from one organization to another is not considered to be a modification. See "TRANSFERRING A DOMAIN NAME".

The Domain Template MODIFY option CAN NOT be used to change the Domain name itself. If you wish to change the Domain you are using, file a separate NEW DOMAIN template followed by a DELETE DOMAIN template when you are ready to have the old name removed.

The Domain Template MODIFY option CAN NOT be used to change the contents of either a Contact or Host record. Use the corresponding Contact Template or Host Template to make these changes. See:

ftp://rs.internic.net/templates/contact-template.txt
ftp://rs.internic.net/templates/host-template.txt

If a Contact of the Domain has previously used the Contact Template to specify the use of Pretty Good Privacy (PGP) or encrypted password, the authentication scheme and information associated with it should be listed in items 0b and 0c. These items contain the authorization information for the sender of an update request and help determine if the request is coming from a valid sender. These items are REQUIRED when a domain is being modified.

```
   Item 0b is the type of authentication scheme.  It can have values
   MAIL-FROM, CRYPT-PW, or PGP.  Item 0c is the authorization information
   for the selected authentication scheme.  As shown by the table below,
   items 0b and 0c are REQUIRED only when CRYPT-PW is the selected
   authentication scheme.

   The different items 0b and 0c combinations are:

   If Item 0b is      Then Item 0c is

   MAIL-FROM          Ignored.  The FROM field in the mail header of
                      an update message will be checked to verify the
                      sender.

                      MAIL-FROM is the DEFAULT authentication scheme.

   CRYPT-PW           Cleartext password (the plain text of the encrypted
                      password).

   PGP                Ignored.  The sender should sign the entire update
                      message with its secret PGP key and send it in
                      cleartext to the InterNIC.


Section 1 - Purpose of Registration

   Briefly describe the purpose of the modification.  If the sender
   is not a currently listed Contact, explain your relationship to
   the current Contacts and/or the organization holding the name.


Section 2 - Complete Domain Name

   Insert the two-part name of the Domain name you wish to modify.
   For example, EXAMPLE.COM.  Item 2 is REQUIRED.


Section 3 - Organization Using the Domain Name

   The Domain name is considered to be registered to an organization,
   even if the organization is an individual.  Therefore, any
   significant change in the organization name or address should be
   explained in Section 1.  Transfer of a name from one organization
   to another is not considered to be a modification.  See instructions
   in "TRANSFERRING A DOMAIN NAME".


Section 4, 5, 6 - Contacts

   A Contact can only be changed by replacing with another individual
   or role Contact.  Replacement can be accomplished by placing the
   handle of a Contact that is already registered in item "a" or by
   registering a new Contact by completing items "b" through "l".

   To modify the record of an existing Contact, use the Contact Template
   available at:

      ftp://rs.internic.net/templates/contact-template.txt


Section 7, 8 - Name Servers
```

When modifying the Name Server list, provide the COMPLETE LIST of Name Servers. The same instructions apply as for creating a new Domain. The Domain Template CAN NOT be used to change the name or IP address of a Name Server that is already registered. To modify the record of an existing Host, use the Host Template available at:

ftp://rs.internic.net/templates/host-template.txt

Section 9 - Invoice Delivery

If you wish to receive your invoice electronically, enter the character E or the word EMAIL in item 9. If you wish to receive your invoice by postal mail, enter the character P or the word POSTAL in item 9.

DELETING A DOMAIN NAME RECORD

A Domain name may be removed from the InterNIC database and hence from the Root Servers under the following conditions:

- A request for deletion comes from an authorized source
- As part of a transfer to another organization
- Failure to provide name service for a 90-day period
- Failure to pay the registration or maintenance fee

If a request for deletion comes for the Administrative or Technical Contact, and meets the security requirements, it will be processed. If the request comes from the Internet Service Provider (ISP) currently providing name service for the Domain name, the name will be placed on HOLD for 60 days to give the organization a chance to find another provider.

If the request comes from none of the above, and does not meet any security established for the Contacts, it will be returned for further explanation of the relationship between the requester and holding organization. The Administrative and Technical Contacts will be notified via email.

When a name is transferred from one organization to another, the InterNIC will first delete the existing record and then process a new registration. See instructions in "TRANSFERRING A DOMAIN NAME".

If an ISP removes name service or if the InterNIC detects lack of name service through its lame delegation policy, the name will be placed on HOLD. HOLD means that the Domain will be visible via WHOIS but will not be active in the Root Servers, that is, it can not be used. The organization holding the name will have up to 60 days to get name service restored. If at the end of this time, active Name Servers have not been provided, the name will be removed. See the lame delegation policy draft available at:

ftp://rs.internic.net/policy/internic/internic-domain-5.txt

Notification of the deletion and the approximate time it will take effect will be sent to:

- the requester
- the current Contacts
- the Technical Contact for the Domain in which the Primary Name Server resides

Section 0 - Authorization

In item 0a, enter the character D or the word DELETE to indicate the deletion of an existing Domain name registration.

If a Contact of the Domain has previously used the Contact Template to specify the use of Pretty Good Privacy (PGP) or encrypted password, the authentication scheme and information associated with it should be listed in items 0b and 0c. These items contain the authorization information for the sender of an update request and help determine if the request is coming from a valid sender. These items are REQUIRED when a domain is being deleted.

Item 0b is the type of authentication scheme. It can have values MAIL-FROM, CRYPT-PW, or PGP. Item 0c is the authorization information for the selected authentication scheme. As shown by the table below, items 0b and 0c are REQUIRED only when CRYPT-PW is the selected authentication scheme.

The different items 0b and 0c combinations are:

If Item 0b is	Then Item 0c is
MAIL-FROM	Ignored. The FROM field in the mail header of an update message will be checked to verify the sender. MAIL-FROM is the DEFAULT authentication scheme.
CRYPT-PW	Cleartext password (the plain text of the encrypted password).
PGP	Ignored. The sender should sign the entire update message with its secret PGP key and send it in cleartext to the InterNIC.

Section 1 - Purpose of Registration

Briefly describe the reason for the deletion. If the sender is not a currently listed Contact, explain your relationship to the current Contacts and/or the organization holding the name.

Section 2 - Complete Domain Name

Insert the two-part name of the Domain name you wish to have deleted. For example, EXAMPLE.COM. Item 2 is REQUIRED.

Section 3 - Section 9

These Sections should be removed or left blank.

TRANSFERRING A DOMAIN NAME

A domain name should be tansferred ONLY if the organization is changing. Two separate templates are REQUIRED to transfer a Domain name. The first template requests a DELETE of the Domain name by the current Domain name holder. The second template requests a NEW Domain name registration from the organization seeking to obtain the Domain name. Whenever possible, these two templates should be sent in the same message.

The following procedures apply:

The Domain name Administrative Contact listed in the Domain name record on file with InterNIC Registration Services MUST complete Sections 0-2 of the Domain Template. Item 0a should reflect DELETE. The subject line of the message should read "TRANSFER DOMAIN" followed by the Domain name being transferred. If the current Contact no longer has access to the Internet, a fax authorizing the transfer may be sent to the InterNIC and the recipient.

Send the DELETE template to the person representing the organization to which the name is to be transferred.

If the recipient of the Domain name does not have Internet access, they can obtain an email account through the Internet Service Provider (ISP) they have selected to support the Domain name.

The recipient of the Domain name should complete a different Domain Template with item 0a reflecting NEW.

Append the second template to the email from the current Domain holder and forward the joined templates as a single message to HOSTMASTER@INTERNIC.NET.

Please note that the email headers from both parties MUST be intact to provide evidence of appropriate authorization. If it is not clear that the transfer is authorized by the current holder, the templates WILL NOT be processed.

D

La facture de l’InterNIC

Dans les sept jours qui suivent une demande d’enregistrement de domaine, l’InterNIC envoie par courrier électronique une facture dont voici un exemple.

```
[ URL ftp://rs.internic.net/templates/sample-invoice.txt ]          [ 10/95 ]

*************************************************************************

                            SAMPLE INVOICE

This is an example invoice that will be sent within seven days after your
registration of your domain with the InterNIC. On the new domain template
there is an option to have your invoice delivered to you either by postal
mail or by electronic mail to the billing contact. If the billing contact
does not exist, the invoice will be delivered to the administrative contact.

You then can remit payment to the InterNIC either by fax (for credit
cards and pre-established accounts only) or by postal mail. The information
is listed below.

*************************************************************************

PAYOR:                                                DATE: 19-sep-1995
John Doe
My Internet Company, Inc.
11 Main Street
Anytown, Anywhere 94402
AnyCountry

PAYEE:
InterNIC Registration Services
P.O. Box 1656
Herndon, VA 22070
US

FAX: +1 (703) 742-4811
E-MAIL: invoice@internic.net
```

```
*************************************************************************

                INVOICE FOR DOMAIN REGISTRATION & RENEWAL

****************** Please DO NOT REMOVE Version Number ******************

Invoice Version Number: 1.0

*************** Please see attached detailed instructions *****************

Invoice Information
1a. Invoice Number.....................: 950915.1
1b. Invoice Date.......................: 15-sep-1995
1c. Invoice Due Date...................: 14-nov-1995

Domain Information
2a. Registered Domain Name.............: DOMAIN.COM
2b. Associated Tracking Number.........: 950829.78
2c. Initial Registration Fee...........: $100.00 USD
2d. (A)ccept (R)eject Charge...........:

Summary Information
3a. Total Charges......................: $100.00 USD
3b. Total Amount Accepted..............:

Payment Method
4a. (CH)eck, (CR)edit card, or (AC)count:

Credit Card Payment Information
5a. (V)isa, (M)astercard, or (A)MEX.....:
5b. Credit Card Number.................:
5c. Name on Credit Card................:
5d. Expiration Date....................:
5e. Authorized Signature and Date.....:

Account Payment Information
6a. Account Number (5-Digit)...........:

-------------------------------- cut here --------------------------------

                           GENERAL INSTRUCTIONS


The form above details the domain name(s) you have registered and offers three
possible payment methods.  Use this form to indicate your acceptance and
payment of domain names.  THIS FORM AND YOUR PAYMENT IN US DOLLARS (USD)
SHOULD BE REMITTED EITHER BY POSTAL MAIL OR BY FAX.  The mailing address
is:

  InterNIC Registration Services
  P.O. Box 1656
  Herndon, VA 22070

The fax number is:

  +1 (703) 742-4811

Please do NOT remit payment information electronically.  Questions concerning
this form should be directed to billing@internic.net.

If this form and your payment are received by the due date, your use of the
domain name(s) will continue.  The invoice due date is shown in item 1c.  You
will receive a reminder message 10 days before the due date if payment has
```

not been received. The payment MUST be received by the due date in order for your use of the domain name(s) to continue. This form MUST accompany your payment in order for payment to be processed, unless you are paying for only one domain by check. In that case, you may simply write the domain name on your check. Your canceled check or credit card statement serves as your receipt.

Please do not modify the form nor remove the version number. The instructions for completing each field follow. Please read the instructions carefully and make sure the form is properly completed to accomplish the action you desire.

Section 1 - Invoice Information

Do not alter these fields. Your payment is due on the date shown in item 1c.

Section 2 - Domain Information

Do not alter items 2a, 2b or 2c. In item 2d, please place the character "A" or the word "ACCEPT" to indicate your acceptance of and intent to pay this domain. If, for any reason, the domain is unacceptable, please indicate such with the character "R" or the word "REJECT." A domain that is "rejected" will be considered unregistered and will be returned to the free pool.

Section 3 - Summary Information

Do not alter item 3a. The sum of the fees associated with those domains you have accepted should be placed in item 3b. This is the amount of your payment.

Section 4 - Payment Method

Indicate the payment method you prefer in item 4. Enter "CK" or the word "CHECK" if paying by check. Enter "CR" or the word "CREDIT" if paying by credit card. Enter "AC" or the word "ACCOUNT" if paying on account.

Section 5 - Credit Card Payment Information

If paying by check or on account, this section should be left blank. If paying by credit card please list the name of the card in item 5a. Please use either a "V" (or the word "Visa"), an "M" (or the word "Mastercard") or an "A" (or the word "AMEX"). List your credit card number in item 5b. The name, as it appears on your credit card, should be written in item 5c. The credit card expiration date should be listed in item 5d. Your signature and date should appear in item 5e.

Section 6 - Account Payment Information

THIS SECTION IS FOR USE BY INTERNET SERVICE PROVIDERS ONLY. Complete item 6 only if you intend to pay on account. To pay on account, you must first complete the Account Application Form available at ftp://rs.internic.net/templates/account-template.txt and submit it with your initial payment. After your Application Form and payment have been received, you will be assigned a 5-digit Account Number. You may then charge your registration fees against your account by writing your account number in item 6.

E

Les fournisseurs Internet

Le choix d'un fournisseur Internet est une opération toujours délicate. En effet, il peut être coûteux et complexe de changer de prestataire :

- par exemple, si le prestataire a pris à sa charge la location d'une ligne spécialisée entre son centre d'opérations et les locaux de son client, ce dernier doit supporter les frais d'une nouvelle mise en place d'une ligne spécialisée avec son nouveau fournisseur, alors que s'il avait été lui-même le client de l'opérateur Télécom, il n'aurait eu qu'à demander un déplacement de la boucle locale d'un fournisseur vers l'autre, le coût de l'opération étant alors plus faible ;

- si c'est le fournisseur qui a attribué une plage d'adresses IP à son client, ce dernier se voit dans l'obligation de renuméroter ses machines lorsqu'il change de fournisseur, opération souvent complexe à planifier.

Pour choisir un fournisseur qui satisfasse à ses propres besoins, il faut se poser un certain nombre de questions :

- **assistance technique** : le fournisseur possède-t-il une assistance technique efficace, la HotLine est-elle payante ou s'agit-t-il d'un numéro vert, quel est le temps moyen d'attente avant de pouvoir contacter un technicien ? L'assistance téléphonique est-elle disponible le week-end, la nuit ? Le fonctionnement du réseau est-il garanti 24h/24 ?

- **qualité des lignes internationales** : le fournisseur possède-t-il ses propres lignes trans-atlantiques, les débits sont-ils suffisants (plusieurs Mb/s, voire même plusieurs dizaines de Mb/s) ? Il faut vérifier que le fournisseur a des accords de *peering* locaux avec les autres fournisseurs du même pays. Par exemple, en France, il existe un point d'inter-connexion de nombreux fournisseurs, nommé le F-GIX (French Global Internet Ex-change).

- **dimensionnement des accès** : il faut vérifier que le fournisseur est correctement dimensionné par rapport au nombre de clients qu'il possède. Dans le cadre d'un accès RNIS, il faut vérifier que le nombre de lignes RNIS du fournisseur est suffisant pour qu'à aucun moment de la journée l'accès ne soit impossible. Dans le cadre d'un accès par modem, il faut vérifier que l'opérateur dispose de sufisamment de modems et qu'il propose des connexions rapides (28800 bits/s, 33600 bits/s). Dans le cadre d'un raccordement à travers un réseau X25, par exemple Transpac en France, il faut vérifier que l'abonnement du fournisseur auprès de l'opérateur X25 est suffisant (64Kb/s, 128Kb/s voire plus).
- **services fournis** : certains fournisseurs proposent de nombreux services en plus de la connexion à l'Internet. Par exemple, il peut s'agir de la mise à disposition de miroirs FTP ou d'un proxy-cache sur le réseau du fournisseur, de liste de distribution de tickets pour le suivi d'incidents, de l'accès aux services multicast par la mise en place de tunnels d'encapsulation, de la prise en charge à distance de l'administration du routeur dans le cadre d'un accès par ligne spécialisée, de la location des routeurs de proximité (RNIS et ligne spécialisée).
- **système de tarification** : les systèmes de tarification des différents fournisseurs sont souvent difficiles à comparer. Certains vont facturer au volume de données transmises, d'autres vont fixer un forfait dépendant du type de raccordement, d'autre encore vont mixer ces deux types de tarification en proposant des tarifs au volume, mais plafonnés. Certains fournisseurs vont jusqu'à différencier les coût des datagrammes IP en fonction de leur destination : France, Europe ou reste du monde. Choisir un fournisseur qui facture en fonction du volume, et mettre en place sur le site un proxy-cache en contrôlant l'utilisation du réseau peut souvent être plus économique que de choisir une tarification forfaitaire.

Le NIC-France met à jour une liste des fournisseurs Internet en France. Celle-ci peut être consultée à l'URL suivant :

```
http://www.nic.fr/Prestataires/index.html
```

Extraits de cette liste, voici les fournisseurs proposant au moins le service de raccordement par ligne spécialisée Transfix :

Nom	**Adresse**
Actif Plurimedia `noc@apm.fr`	22, bld des tchecoslovaques 69007 Lyon
Activnet `jbernex@dtr.fr`	27, rue de la Villette 69003 Lyon

Aic Technologie info@aic.fr	8, rue Benoit Frachon 38090 Villefontaine
Alpes Networks seb@alpes-net.fr	11, rue Hebert 38000 Grenoble
Archimedia	Centre Condorcet Rue du Docteur Schweitzer 33600 Pessac
Artinternet noc@argia.fr	16-24, rue Louis-Pasteur 92100 Boulogne-Billancourt
ASI postmaster@asi.fr	Espace Double Mixte 43, bd du 11 Novembre 1968 69622 Villeurbanne Cedex
Axnet webmaster@axnet.fr	rue de la chermerie 21120 Spoy
Bull/IBT info@bull.fr	International Bull Telecommunications 20, rue Dieumegard 93406 Saint-Ouen
Cadrus info@cadrus.fr	Parc Technologique du Canal 13, av de l'Europe 31527 Ramonville-Saint-Agne
Centre internet Europeen info@cie.fr	6, rue Martel 75010 Paris
Celya noc@celya.fr	9, rue Alfred Kustler BP 60762 Nantes Cedex 3
CGE Online r.scultore@cge-ol.fr	33 ter, rue de la Figairasse 34070 Montpellier
Computers & Communications sales@magicdom.com	75000 Paris
Crdi Systèmes crdi@crdi.fr	36, rue Feydel 46000 Cahors
CyberAccess access@cyberaccess.fr	Parc des Glaisins 14, rue du Pré Paillard 74940 Annecy le Vieux
Dotcom info@dotcom.fr	124, boulevard de Verdun 92400 Courbevoie
DX Net info@dx-net.fr	21, rue des Bosquets 67300 Schiltigheim
Easynet info@easynet.fr	23, rue du Renard 75004 Paris

EBC webmaster@ebc.net	109, rue Edmond Rostand 51100 Reims
EUnet France contact@EUnet.fr	52, Avenue de la Grande Armée 75017 Paris
EuroTeleport de Roubaix pbouffel@etnet.fr	84, boulevard du Général Leclerc 59100 Roubaix
Franche Comté Net info@mail.fc-net.fr	1, rue Gay Lussac 25000 Besançon
Free-Net France p.tordjman@freenet.fr	26, rue Bancel 77000 Melun
Groupe CX info@iway.fr	358, chemin Hugues Berenguier 06610 La Gaude
Grolier France info@grolier.fr	131, av. Charles de Gaulle 92200 Neuilly-sur-Seine
GSI infos.NCS@gsicom.com	4, rue Sentou 92150 Suresnes
Icor infos@icor.fr	228, rue Paul Gidon Z.I. de Bissy 73000 Chambéry
iLink fti@wanadoo.com	41, rue Camille Desmoulins 92442 Issy-les-Moulineaux
Imaginet vente@imaginet.fr	21, rue de la fontaine au roi 75011 Paris
Infoshop contact@infoshop.fr	40, rue de l'abbé Lemire 59110 La Madeleine
Infrescom infos@infrescom.fr	12, rue du Moulin 25150 Vermondans
In'Net info@inba.fr	Aquitaine Chauveau 33420 Espiet
Internet - Communications tech@inlandsys.com	6, rue Joule Mérignac Phare 33700 Mérignac
Internet-Plus info@iplus.fr	19, rue Réaumur 75003 Paris
Internet Solutions ydoffin@starnet.fr	173, rue de Vaugirard 75015 Paris
Internet Way info@iway.fr	204, boulevard Bineau 92200 Neuilly
Internext info@internext.fr	50, bd du Colonel Fabien 94200 Ivry-sur-Seine

Internix internix@slnet.mc	13, avenue Des Papalins BP 604 98013 Monaco
Jovenet www@jovenet.fr	15, rue Peyras 31000 Toulouse
Ludexpress e.etcheparre@finest.tm.fr	32, Cours Alsace Lorraine 33000 Bordeaux
Macorbur-Internext info@inext.fr	56, rue Paul Claudel 87000 Limoges
Mnet webmaster@www.mnet.fr	Parc Technologique de la Pompignane Rue de la vieille poste 34055 Montpellier Cedex 1
NCTech info@nctech.fr	8, rue Hermann Frenkel 69007 Lyon
Nucleus Informatique info@nucleus.fr	8, rue de Vieux-Thann 68200 Mulhouse
OLEANE info@oleane.net	Les Collines de l'Arche Opera C 92057 Paris La Defense
Pictime info@pictime.fr	121, rue de Chanzy 59260 Hellemmes-Lille
Planete Net/Pressicom info@planete.net	5/7, rue Raspail 93108 Montreuil Cedex
Promethee tech@promethee.fr	5, rue du Plan d'Agde 34000 Montpellier
Pyrenet ohp@pyrenet.fr	6, chemin d'Harraud Turrau 31190 Auterive
Quaternet info@quaternet.fr	av JF Kennedy 33700 Mérignac
Remcomp info@remcomp.fr	93, Rue du Faubourg Saint Martin 75010 Paris
Renater rensvp@renater.fr	Université Pierre et Marie Curie 4, place Jussieu 75252 Paris Cedex 5
Sdv Plurimedia info@sdv.fr	15, rue de la nuee bleue 67000 Strasbourg
Skyworld info@sky.fr	98, rue Barrault 75013 Paris
Smart on Line servcoml@smartonline.fr	Les Brosses 53440 La Bazoge

Sprint `webmaster@sprintlink.net`	164, bis avenue Charles De Gaulle 92522 Neuilly cedex
The Pandemonium Group	20, rue du Maréchal Foch 67380 Lingolsheim
Topnet	Tech'Indus C 150, rue Mayor de Montricher Aix-en-Provence Cedex 3
Transeo `info@transeo.fr`	8, rue du Delta 75009 Paris
Transpac/RAIN `noc@rain.fr`	12 bis, rue Campagne Première 75014 Paris
Unisoft `info@unisoft.fr`	23, rue Gabriel Péri 31000 Toulouse
Uplift Technology `info@uplift.fr`	18, rue Pradier 75019 Paris
Vtcom `vtcom@vtcom.fr`	40, rue Gabriel Crie 92240 Malakoff

Codes d’état du protocole http

100	Continue
101	Switching Protocols

200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content

300	Multiple Choices
301	Moved Permanently
302	Moved Temporarily
303	See Other
304	Not Modified
305	Use Proxy

400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Time-out
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request URI Too Large
415	Unsupported Media Type

500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-out
505	HTTP Version not supported

G

Compression et archivage

G.1 Généralités

Il est parfois indispensable de se livrer à une série de manipulations avant d'obtenir un fichier prêt à être envoyé. L'utilisation d'un système d'encodage augmente la taille des fichiers qu'il devient donc nécessaire de compresser avant l'encodage pour limiter le plus possible cette inflation. Les fichiers à transférer peuvent être nombreux et de petite taille; archivés dans un fichier unique, ils se compresseront mieux, arriveront plus rapidement et seront plus commodes à envoyer. Les informations propres au système (comme les répertoires, les dates de mise à jour des fichiers, ou encore leur propriétaire et les droits d'accès qui y sont associés) ne peuvent être expédiées que si les fichiers sont archivés et si l'archive répertorie ces informations.

Pour simplifier les transferts, on utilise donc principalement trois techniques (dans cet ordre) :

- *l'archivage*, qui fabrique un fichier unique à partir de plusieurs fichiers en y intégrant les éléments nécessaires pour reconstituer les fichiers après le transfert ;
- *la compression*, qui réduit la taille du fichier ou de l'archive ;
- *l'encodage*, dont nous avons déjà parlé, qui assure que le transfert du fichier se passera bien et qu'aucune information ne sera perdue à cause d'équipements ou de logiciels qui ne laisseraient pas passer certains codes.

De nombreux utilitaires regroupent en une seule étape les fonctions d'archivage et de compression. Les plus utilisés sont `PKZIP`, `LHARC` ou `ARJ` du côté PC, `STUFFIT` du côté Mac, `zip` (l'équivalent de `PKZIP`, dont il comprend d'ailleurs le format) ou `lha` (la version Unix de `LHARC`) pour les systèmes de type Unix.

Hérités des systèmes Unix, les utilitaires `tar` et `gzip` ou `compress` nécessitent quant à eux

deux étapes, `tar` assurant la fonction d'archivage et `gzip` ou `compress` la compression de l'archive. Leur avantage sur d'autres utilitaires est qu'ils permettent un meilleur contrôle des informations propres au système et une compression souvent plus performante. Ce sont en outre les outils les plus couramment utilisés sur l'Internet, car ils établissent un standard de facto.

Ils sont disponibles sur presque tous les systèmes, y compris MS-DOS/Windows et OS/2, ce qui n'est pas le cas de tous les autres (les formats spécifiques aux systèmes Mac étant les plus délicats à traiter sur d'autres plates-formes).

G.2 Archivage et compression sous Unix

G.2.1 Tar

L'utilitaire `tar` est disponible sur toutes les plates-formes de type Unix. Conçu à l'origine pour archiver des arborescences complètes ou des groupes de fichiers sur des supports externes (cartouches, bandes...), il est désormais très utilisé pour fabriquer des *distributions* de programmes diffusés sur l'Internet. Il inclut des fonctions avancées de mise à jour différentielle qui ne font pas l'objet de cette annexe.

Bien que l'extension `.tar` n'ait aucun caractère impératif, elle est très largement répandue. Le *désarchivage* d'un fichier fabriqué avec `tar` s'effectue en spécifiant les options `x` (pour eXtraire) et `f` (pour Fichier). Le nom de fichier est indiqué après l'option `f` qui doit être la dernière.

```
luc$ tar -xf archive.tar
```

L'option `v` (*verbose*) permet d'obtenir la liste des fichiers au fur et à mesure de leur extraction. Dans la plupart des cas simples, le tiret commun aux options des commandes sous Unix n'est pas nécessaire.

```
luc$ tar xvf archive.tar

./demain/travail.a.faire
./demain/travail.a.faire.faire
./demain/travail.a.reporter
./hier/travail.non.fait
luc$
```

Pour visualiser simplement le contenu de l'archive (par exemple avant de lancer l'extraction), il suffit de remplacer l'option `x` par `t`. Puisque `tar` archive des arborescences complètes ou des parties d'arborescence, les chemins d'accès sont également archivés. Le plus souvent, ils sont enregistrés de manière relative, c'est-à-dire par rapport au répertoire courant. Ainsi, l'extraction placera les répertoires directement sous le répertoire depuis lequel elle sera lancée.

L'option `c` permet de créer une nouvelle archive. Le nom de l'archive à créer est suivi de la liste des répertoires ou fichiers à archiver. Les répertoires spécifiés seront archivés complètement, de manière récursive (c'est-à-dire avec leurs sous-répertoires). Pour fabriquer une

archive dans laquelle les répertoires sont indiqués par rapport à la racine de l'arborescence, il suffit d'utiliser l'option `P` ; cette option n'est pas recommandée, sauf si le contenu de l'archive doit impérativement être installé dans un répertoire particulier.

Exemples

`tar xf archive.tar` extrait l'arborescence et les fichiers contenus dans le fichier `archive.tar`

`tar tf -` liste le contenu de l'archive lue sur l'entrée standard

`tar cvf nouvelle_archive.tar .` archive le contenu du répertoire courant dans le fichier `nouvelle_archive.tar`

`tar rPvf archive.tar *.gif` ajoute tous les fichiers GIF du répertoire courant à la fin du fichier `archive.tar`, en affichant leur nom et en conservant leur chemin d'accès complet

G.2.2 Principales options de GNU tar

Certaines de ces options sont spécifiques à la version GNU de `tar` et ne figurent pas dans d'autres versions (cas par exemple des options `-z` et `-Z`). `tar` attend une (et une seule) des options suivantes :

`-A`	Concaténer deux archives.
`-c`	Créer une nouvelle archive.
`-d`	Comparer l'arborescence des fichiers contenus dans une archive à une arborescence sur le disque.
`-r`	Ajouter des fichiers à la fin d'une archive. Attention, les fichiers sont ajoutés même si une version précédente se trouve déjà dans l'archive. Outre le fait qu'à l'extraction, sauf manipulation particulière, le fichier obtenu en fin de compte sera le dernier fichier archivé, cette option peut également présenter l'inconvénient de faire grossir démesurément l'archive.
`-u`	Ajouter des fichiers à la fin d'une archive, mais uniquement s'ils sont plus récents que les versions déjà présentes (où s'ils n'y sont pas déjà). Voir l'option `-r`.
`-x`	Extraire les fichiers contenus dans une archive. Pour n'extraire qu'une partie des fichiers, il faut spécifier leur nom sur la ligne de commande.

Lorsqu'on utilise `tar` pour fabriquer des archives afin, par exemple, de transmettre des fichiers par FTP, il est toujours nécessaire de préciser un nom de fichier archive (dans l'absolu, ce n'est pas indispensable : en l'absence de nom de fichier, `tar` tente de lire ou d'écrire sur un lecteur de bande).

Ce nom de fichier doit suivre immédiatement l'option `-f`. On peut en outre ajouter ces options facultatives (la liste n'est pas complète) :

`-h`	Lors de la création d'une archive, les liens symboliques sont enregistrés comme tels. Cette option permet de demander qu'ils soient remplacés par les fichiers originaux. Il faut l'utiliser dès lors que certains fichiers seraient manquants si on n'archivait que les liens, mais également si l'archive est destinée à être extraite sur un système qui ne connaît pas la notion de lien symbolique (Windows, par exemple). Attention toutefois à l'augmentation de la taille de l'archive : évitez de dupliquer les fichiers.
`-k`	Lors d'une extraction, les fichiers existants sont éventuellement écrasés par les fichiers extraits. Cette option garantit qu'aucun fichier existant ne sera remplacé.
`-p`	Cette option permet d'extraire les fichiers d'une archive en conservant les droits d'accès originaux.
`-P`	Lors de l'archivage, les chemins d'accès enregistrés sont relatifs au répertoire courant. Ainsi, lorsque vous créez une archive avec la commande `tar cf archive.tar /usr` `tar` supprime le premier `/`. Lors de l'extraction, par exemple sous le répertoire `/home/extract`, les fichiers sont placés dans `/home/extract/usr`. L'option `-P` indique à `tar` qu'il ne faut pas supprimer le premier `/` s'il est spécifié. Elle permet d'archiver des fichiers qui doivent impérativement être extraits dans un répertoire particulier (par exemple les fichiers spéciaux du répertoire `/dev`).
`-T`	Cette option, utilisée en création ou mise à jour, doit être immédiatement suivie d'un nom de fichier contenant la liste des fichiers à ajouter à une archive (un nom de fichier par ligne).
`-v`	Cette option permet d'obtenir plus d'informations sur l'exécution du programme. Lors d'une extraction ou d'une création, les noms de tous les fichiers et répertoires traités sont affichés. Avec l'option `-t`, la liste des fichiers comporte également les informations du système (celles obtenues avec un `ls -l`) : nom et groupe du propriétaire, droits d'accès et date de dernière modification.
`-w`	Cette option indique à *tar* qu'il doit demander confirmation avant chaque action (qu'il s'agisse d'une mise à jour, d'un ajout de fichier, ou encore d'une création de fichier lors d'une extraction...).
`-X`	Inverse de `-T`, cette option doit être suivie d'un nom de fichier contenant la liste des fichiers qui ne doivent *pas* être ajoutés à l'archive. Notons que les deux options, `-T` et `-X`, ne peuvent pas être utilisées ensemble (l'option `-X` ne serait pas prise en compte).

`-Z`, `-z`	Cette option indique à `tar` qu'il doit appliquer un programme de compression, `compress` (`-Z`) ou `gzip` (`-z`) lors de l'extraction ou de la création.

G.2.3 Gzip et compress

Ces deux utilitaires, auxquels on peut associer leurs 'inverses' `gunzip` et `uncompress` (ce sont en réalité les mêmes programmes, représentés par des liens symboliques vers `gzip` et `compress`, leur fonction étant déterminée par le nom sous lequel ils sont appelés), compressent un fichier en appliquant un algorithme comparable à celui utilisé dans la compression des images au format GIF ou dans la compression mise en oeuvre par les modems (V42bis).

Schématiquement, ce type de compression, dit de *Lempel-Ziv*, utilise un dictionnaire, une table recensant des séquences qui ont tendance à se répéter souvent, et code ces séquences sous la forme d'un index, dont la longueur (en octets ou en bits) est moindre que celle de la séquence vers laquelle il pointe. Ainsi, lorsque dans un document le mot pomme est utilisé plusieurs fois, il sera probablement inséré dans le dictionnaire et remplacé dans le texte par une référence numérique.

Bien sûr, la performance de l'algorithme réside dans sa faculté de coder les chaînes les plus longues possibles par des références les plus courtes possibles, tout en maintenant un compromis acceptable entre la compression obtenue et le temps nécessaire au traitement. Les facteurs de compression habituels se situent autour de 40%, souvent plus si le fichier compressé contient beaucoup de texte (jusqu'à 90% pour un document PostScript, dont la syntaxe est généralement très redondante), et moins s'il s'agit d'une image (les photos numérisées en 16 millions de couleurs se compressent très mal avec cette technique, car elles contiennent peu d'informations redondantes; la réduction des couleurs permet d'augmenter cette redondance, ce qui explique les performances affichées par la compression LZW utilisée dans le format GIF).

L'utilisation de `compress` et `uncompress` est des plus simples. Ces commandes acceptent en paramètre un ou plusieurs noms de fichiers à compresser ou décompresser.

Après compression, le fichier `archive.tar` sera remplacé par le fichier `archive.tar.Z`. Inversement, `uncompress` transformera `archive.tar.Z` en `archive.tar`. Notons que l'extension `.Z` est imposée, sans quoi le programme refuse de décompresser le fichier, même s'il est au bon format.

L'option `-c` permet de rediriger le résultat d'`uncompress` vers la sortie standard au lieu de remplacer le fichier compressé (la commande `zcat` réalise le même travail), ce qui permet de consulter le début d'un fichier avant de choisir s'il faut ou non le décompresser.

La commande `gzip` fonctionne d'une manière tout à fait similaire. Elle offre cependant une grande richesse d'options supplémentaires. Les plus courantes sont `-1` et `-9` qui permettent de choisir entre une compression rapide mais moins efficace et la meilleure compression pos-

sible au détriment du temps. L'option `-d` permet par ailleurs de demander la décompression du fichier, ainsi `gzip -d` et `gunzip` sont équivalents.

```
gide: $ ls apache.1.1/
CHANGES acl/ gd1.2/ icons/ php/ src/
LICENSE cgi-bin/ htdocs/ inetd/ php-2.0b2/ support/
README conf/ httpd@ logs/ php_fi/
gide: $ tar cf apache.1.1.tar apache.1.1
gide: $ gzip -9 apache.1.1.tar
gide: $ ls -l apache.1.1.tar.gz
-rw-r--r-- 1 www wwwadm 1382400 Aug 28 16:28 apache.1.1.tar.gz
```

Figure G.1 *Exemple de fabrication d'une archive compressée*

H

Contenu du CD-ROM

H.1 Conditions d'utilisation du CD-ROM

La plupart des logiciels fournis sur le CD-ROM d'accompagnement sont des logiciels du domaine public, qui peuvent être utilisés gratuitement dans le cadre fixé par la notice de copyright ou licence d'utilisation qui les accompagne. Le logiciel Netscape Navigator 3.0, fourni sur le CD-ROM en versions Windows, Macintosh et Unix, fait l'objet d'une licence d'utilisation particulière, dont les termes sont détaillés dans la section H.5.

Tous les efforts ont été faits pour tester avec soin ce CD-ROM et les programmes qu'il contient. Néanmoins, les éditions Eyrolles ne sauraient être tenues pour responsables des préjudices ou dommages de quelque nature que ce soit, pouvant résulter d'un mauvais fonctionnement ou d'une mauvaise utilisation de ces programmes.

Si le lecteur éprouve des difficultés à l'utilisation ou à l'installation d'un de ces utilitaires, il pourra adresser ses questions dans des forums Internet tels que :

- `fr.network.divers`,
- `fr.comp.sys.mac`,
- `fr.comp.sys.pc`,
- `fr.comp.os.unix`.

Il est vivement recommandé de parcourir le forum `news.announce.newusers`, destiné à fournir toutes les informations nécessaires à l'utilisation du système de forums.

H.2 Unix

H.2.1 Logiciels en version binaire

Le CD-ROM contient les distributions binaires des logiciels suivants, destinés à l'environnement Unix :

- **Netscape Navigator 3.0 pour plates-formes AIX, BSD/386, HP-UX, Irix, Linux, Unix OSF-1, SunOS 4.1.3, Solaris 2.3 et 2.4** : un navigateur permettant de se connecter aux serveurs WWW, FTP, Gopher, d'émettre et de consulter du courrier et de participer aux forums,
- **RAT pour plates-formes Irix 5.3, Linux, HP-UX, SunOS 4.1.3_U1, Solaris 2.4 et 2.5** : un gestionnaire de conférences audio.

H.2.2 Logiciels en version source

Le CD-ROM contient les distributions des sources des logiciels suivants, destinés à l'environnement Unix :

- **ppp-2.2** : une passerelle PPP logicielle,
- **diald 0.14** : un composeur automatique pour passerelle PPP,
- **BIND 4.9.4** : le serveur de noms de Berkeley,
- **dig 2.0** : un utilitaire d'interrogation de DNS,
- **dnswalk 1.8.3** : un utilitaire d'analyse du contenu d'un DNS,
- **lamer** : un utilitaire d'analyse des fichiers journaux d'un DNS,
- **sendmail 8.8.3** : une passerelle SMTP,
- **INN 1.4** : un serveur de forums,
- **cops 1.04** : un analyseur de sécurité système,
- **crack 4.1** : un analyseur de mots de passe,
- **ISS 1.21** : un analyseur de sécurité réseau,
- **SATAN 1.1.1** : un analyseur de sécurité réseau,
- **TCP-Wrapper 7.2** : un contrôleur d'accès,
- **xinetd 2.1.4** : un contrôleur d'accès.
- **cfengine 1.3.16** : un outil complexe mais puissant destiné à faciliter l'administration des réseaux TCP/IP héthérogènes,
- **gzip 1.2.4** : l'utilitaire de compression de GNU, en version source,
- **recode 3.4** : un utilitaire permettant la conversion des jeux de caractères les plus répandus,
- **tar 1.11.8** : la version GNU de l'utilitaire d'archivage,

- **uucp 1.06.1** : une passerelle pour l'accès au réseau UUCP (en deux archives, source et documentation),
- **lynx 2.6** : un navigateur HTTP, FTP et Gopher, en mode texte,
- **Spinner 1.0b12** : un serveur HTTP orienté objet,
- **Apache 1.1.1 et 1.2b2** : un des serveurs HTTP les plus utilisés, ainsi que les versions PostScript (`.ps`) et Acrobat (`.pdf`) de son manuel.

La plupart de ces outils fournis en version source sont destinés à un grand nombre de systèmes Unix distincts. L'utilisateur doit disposer d'un compilateur C afin de pouvoir créer les versions exécutables.

H.2.3 Licenses particulières

Certains de ces outils sont fournis avec des licenses particulières, demandant de faire apparaître les notices suivantes :

- Université de Berkeley (logiciels `sendmail` et `dig`) :

  ```
  This product includes software developed by the University
  of California, Berkeley and its contributors.
  ```

- Collège de Londres (logiciel `RAT`) :

  ```
  This product includes software developed by the Computer Science
  Department at University College London.
  ```

- Groupe Apache (serveur `Apache`) :

  ```
  This product includes software developed by the Apache Group for
  use in the Apache HTTP server project (http://www.apache.org/).
  ```

H.3 Windows 3.1 et Windows 95

Le CD-ROM contient les logiciels suivants, destinés aux environnements Windows 3.1 et Windows 95 :

- **Netscape Navigator 3.0 (versions 16 bits et 32 bits)** : un navigateur permettant de se connecter aux serveurs WWW, FTP, Gopher, d'émettre et de consulter du courrier et de participer aux forums,
- **Eudora Light (versions 16 bits et 32 bits)** : un gestionnaire de courrier électronique,
- **gzip** : l'utilitaire de compression de fichiers de GNU.

H.4 Macintosh

Le CD-ROM contient les logiciels suivants, destinés à l'environnement Macintosh :

- **Fetch 3.0** : un utilitaire d'échange de fichiers par FTP,
- **Netscape Navigator 3.0** : un navigateur permettant de se connecter aux serveurs WWW, FTP, Gopher, d'émettre et de consulter du courrier et de participer aux forums,
- **Eudora Light** : un gestionnaire de courrier électronique,
- **MacPPP 1.1.3 et 2.0.1** : une passerelle PPP logicielle.

Certains de ces logiciels sont fournis au format BinHex (extension `.hqx`), bien connu dans le monde Macintosh. L'utilisateur devra être muni d'un décodeur du type BinHex 4.0 ou Stuffit pour pouvoir les utiliser.

H.5 Netscape Navigator

Les versions de Netscape Navigator 3.0 fournies sur ce CD-ROM sont des versions d'évaluation que vous pouvez utiliser librement pendant 90 jours.

Si vous souhaitez continuer à utiliser Netscape Navigator 3.0 au-delà de cette période d'essai, vous devez en acquérir la licence auprès d'un des distributeurs officiels de Netscape :

France :

Softway
32, Avenue de l'Europe
78140 Vélizy
Tél : 01 39 26 50 00

Access Graphics
8-10, rue de la ferme
92100 Boulogne
Tél : 01 46 10 49 40

Azlan
25 Quai Galliéni
92150 Suresnes
Tél : 01 41 38 15 15

SIT France
9, rue Pierre Poli
92130 Issy les Moulineaux
Tél : 01 41 46 19 30

TOP LOG Interquad
Les Rives de Bagatelles
7, allée de l'ancien pont
92150 Suresnes
Tél : 01 41 18 35 00

Belgique :

Comsol
Koningin Av. Reine Astrid 59A
101780 Wemmel
Tél : 32 2 461 01 70

Suisse :

Instrumatic
Hertistr. 29
8304 Wallisellen
Tél : 41 18 77 37 37

VTX Services
Pré de la tour 10
1009 Pully
Tél : 41 21 721 11 11

Pour les autres pays, veuillez consulter le site Web de Netscape (`www.netscape.com`), sur lequel vous trouverez aussi des informations complètes sur l'ensemble de la gamme de produits Internet proposée par Netscape.

NOTE : aucun support technique n'est fourni par Netscape sur les versions d'évaluation incluses sur ce CD-ROM.

H.6 Eudora Light

Le logiciel Eudora Light de la société Qualcomm, présent sur ce CD-ROM, est une version de démonstration limitée du produit nommé **Eudora Pro 3.0**, pour les plates-formes Windows et Macintosh. Ce dernier est muni de très nombreuses options et facilités qui ne sont pas présentes dans la version Eudora Light. Par exemple, on trouve de puissantes facilités de tri automatique des courriers en fonction de nombreux paramètres dans Eudora Pro 3.0.

De nombreux renseignements sur ces produits ainsi que sur Qualcomm sont disponibles à l'URL `http://www.eudora.com`.

Index

Alexandre Fenyö - Frédéric Le Guern - Samuel Tardieu

Raccorder son réseau d'entreprise à l'Internet

Cet ouvrage de référence présente l'ensemble des techniques nécessaires au raccordement d'un réseau local à l'Internet et à la mise en œuvre d'applications Internet/intranet sur ce réseau.

L'exploration de ces techniques est organisée en **cinq étapes :**

- Les différents types de raccordement (modem, RNIS, ligne spécialisée, X.25) et les protocoles mis en jeu (TCP/IP, PPP…).
- Configuration d'une messagerie SMTP, d'un serveur DNS et d'un serveur de news.
- Techniques WWW : protocole HTTP, langage HTML, interface CGI, serveur proxy, installation de Netscape Commerce Server et de Apache.
- Services IP Multicast : audio et vidéo-conférence, outils de travail coopératif.
- Topologie d'un réseau sécurisé, mise en place d'un firewall (pare-feu), cryptographie et filtrage.

Cet ouvrage s'adresse au professionnel qui doit mettre en place un accès Internet, ou qui dispose déjà d'un tel accès et désire proposer de nouveaux services à ses utilisateurs. Il intéressera aussi tous les étudiants et informaticiens souhaitant acquérir une vision globale du fonctionnement de l'Internet, depuis les protocoles jusqu'aux réalités de l'implémentation.

Sur le **CD-Rom** offert avec ce livre :

- La panoplie complète des logiciels Unix nécessaires à la mise en place d'un serveur Internet/intranet : serveur DNS *(BIND)*, passerelle de messagerie *(sendmail)*, serveur de news *(INN)*, serveurs Web *(Apache, Spinner)*, outils de gestion de la sécurité *(Satan, ISS 1.21…)*, etc.
- Une sélection de logiciels clients : Netscape Navigator 3.0 pour Macintosh, Windows 3.1/95 et huit plates-formes Unix *(versions d'évaluation limitées à 90 jours d'utilisation)*, Eudora Light pour Windows et Macintosh, etc.

Les auteurs sont tous trois diplômés de l'École Nationale Supérieure des Télécommunications de Paris (promos 1993 et 1994). Ils sont responsables de l'animation de cours et de séminaires Internet et multimédia au sein du département Formation de SUN Microsystems.

Alexandre Fenyö *a été membre de l'équipe technique d'EUnet France, un des principaux fournisseurs d'accès à l'Internet en France. Il est actuellement étudiant-chercheur à l'université Paris 6 dans le domaine des réseaux à très haut débit.*

Frédéric Le Guern *est spécialisé dans l'intelligence artificielle et les sciences cognitives. Cofondateur et directeur associé de la société Intuisys, dont les activités sont centrées sur le marketing des services en ligne, il exerce également une activité de conseil pour la mise en place de serveurs Web et d'intranets.*

Samuel Tardieu *est actuellement étudiant-chercheur au département Informatique de l'ENST dans le domaine des systèmes répartis temps-réel et s'occupe activement de sécurité des systèmes informatiques. Il exerce des activités de conseil notamment dans le domaine des systèmes distribués.*

ISBN : 2-212-08951-1

9 782212 089516

NETSCAPE NAVIGATOR™ INCLUDED

Eyrolles