Université de Nice Sophia-Antipolis

Introduction à l'utilisation des réseaux informatiques

Richard Grin

19 décembre 1997

Table des matières

1 Généralités sur les réseaux
1.1 Les différents types de réseaux
1.1.1 Étendue des réseaux
1.1.2 Modalités de transmission des informations
1.1.3 Topologie
1.1.4 Protocole
1.2 Le modèle OSI pour les réseaux
1.2.1 Architecture de réseau en couche
1.2.2 Couches du modèle OSI
1.2.3 Chemin physique et chemin logique

2 Réseaux Ethernet

3 Réseau Internet, TCP/IP 1
3.1 Quelques généralités sur le réseau Internet 1
3.2 Principes généraux de fonctionnement d'Internet 1
3.3 Protocole IP, adresses IP 1
3.4 Protocole TCP 1
3.5 Protocole UDP 1
3.6 Protocole ICMP 1
3.7 Activité réseau sur une machine (netstat) 1

4 Commandes au-dessus de TCP/IP 1
4.1 Généralités 1
4.2 Principes de fonctionnement des applications TCP/IP 1
4.3 Format des adresses Internet 1
4.4 Condition nécessaire pour les autorisations d'accès 1
4.5 Accorder sa confiance à des utilisateurs sur d'autres machines 1
4.5.1 Groupes du réseau 1
4.5.2 Fichiers "rhosts" 1
4.6 Connexions à distance 1
4.6.1 Entre 2 systèmes Unix compatibles (rlogin) 1
4.6.2 Entre 2 systèmes d'exploitation différents (telnet) 2

4.7 Exécution d'une commande Unix distante (rsh) . . . 20
4.8 Copie de fichiers distants . . . 22
4.8.1 Entre 2 systèmes Unix compatibles (rcp) . . . 22
4.8.2 Entre 2 systèmes d'exploitation différents (ftp) . . . 23
4.8.3 Comparaison de rcp et de ftp . . . 28
4.8.4 Traitement des fichiers récupérés . . . 28
4.8.5 Autres moyens de copier des fichiers distants . . . 29
4.9 Informations sur le réseau . . . 30
4.9.1 Utilisateurs du réseau (finger, rwho) . . . 30
4.9.2 Machines du réseau (ping, ruptime) . . . 31
4.10 Chemin suivi par les messages traceroute) . . . 31

5 Services pour l'utilisateur final 32
5.1 Courrier électronique . . . 32
5.2 Conversation sur le réseau . . . 33
5.2.1 Conversation à deux : talk . . . 33
5.2.2 Conversation à plusieurs : IRC . . . 33
5.3 Recherche d'informations sur le réseau Internet . . . 34
5.3.1 Archie . . . 34
5.3.2 Gopher . . . 35
5.3.3 Wais . . . 36
5.3.4 WWW (ou W3) . . . 37
5.4 News . . . 38
5.5 Accès aux bases de données . . . 40
5.6 Listes de diffusion (ou de messagerie) . . . 40
5.7 Stratégie pour une recherche sur le Web . . . 41
5.7.1 Recherche géographique . . . 42
5.7.2 Recherche par thème . . . 42
5.7.3 Recherche par mot-clé . . . 43
5.7.4 Recherche d'une adresse électronique . . . 43
5.7.5 Serveurs spécialisés dans la distribution de logiciels . . . 43
5.8 Récupérer des sites Web . . . 44

6 Extensions du système qui utilisent le réseau 45
6.1 Systèmes de fichiers distribués : NFS, RFS, NIS . . . 45
6.2 Interface graphique : X Window . . . 46
6.2.1 Présentation de X Window . . . 46
6.2.2 Terminaux X - stations de travail . . . 46

7 Réseau Archipel de l'université 48
7.1 Description du réseau . . . 48
7.2 Réseau local de Valrose . . . 48
7.3 Connexions avec l'extérieur . . . 49

7.3.1 Réseaux régionaux ou nationaux . . . 4
7.3.2 Réseaux internationaux . . . 4
7.3.3 Accès à Archipel via un minitel ou un modem . . . 4

Chapitre 1

Généralités sur les réseaux

1.1 Les différents types de réseaux

1.1.1 Étendue des réseaux

Les réseaux informatiques permettent de relier des ordinateurs et des périphériques entre eux.

Les *réseaux locaux* (*Local Area Networks*, LAN, en anglais) relient les matériels informatiques d'un bâtiment, d'une entreprise ou d'une université. Les ordinateurs doivent être assez proches les uns des autres (quelques kilomètres au maximum). Le support des communications est la paire torsadée (ligne téléphonique), le câble coaxial ou la fibre optique.

D'autres réseaux, dits *étendus* (*Wide Area Networks*, WAN, en anglais), permettent de relier des ordinateurs distants de plusieurs milliers de kilomètres. Les communications peuvent utiliser des câbles, des satellites, des lignes dédiées ou passer par des réseaux publics de communication comme TRANSPAC en France.

1.1.2 Modalités de transmission des informations

Si on se place au niveau des connexions entre ordinateurs du réseau, il existe plusieurs types de réseaux :

- les réseaux dans lesquels tous les ordinateurs sont reliés entre eux par des *lignes point à point*. Un ordinateur ne peut communiquer qu'avec un ordinateur avec lequel il est directement connecté. Ce type de réseau ne convient que pour des réseaux très simples ne contenant que très peu d'ordinateurs.

- les *réseaux à diffusion* (*broadcast* en anglais) dans lesquels les messages sont envoyés sur tout le réseau. Celui à qui le message est destiné le capte au passage. On retrouve ce type de fonctionnement pour la diffusion d'une émission de radio ou de télévision. On verra que les réseaux locaux de type Ethernet sont de ce type.

- les réseaux dans lesquels les communications entre ordinateurs sont effectuées au tra vers d'un *réseau de transmission* plus ou moins complexe. Chaque ordinateur n'e relié qu'à un nombre restreint d'autres ordinateurs. Un ordinateur qui veut commu niquer avec un autre envoie son message dans ce réseau de transmission en donnan l'adresse de son destinataire. Le réseau s'arrange pour envoyer le message à son de tinataire en faisant passer le message par un certain nombre d'ordinateurs du réseau La plupart des réseaux à longue distance, et en particulier le réseau Internet, fon tionnent sur ce principe. C'est aussi le type de connexion utilisé par le téléphone.

Les réseaux de transmission se distinguent par la technique qui leur permet de tran mettre les informations entre les machines :

- dans la technique de *commutation de circuit*, un circuit est établi dès le début entr les deux extrémités de la communication. Il est conservé jusqu'à la fin de la commu nication. Cette technique est utilisée pour le téléphone et par les réseaux dits de typ X.21.

- dans la technique de *commutation de messages* (appelée aussi *store-and-forward* e anglais), l'expéditeur d'un message le remet au réseau qui se charge de le faire parven de nœud en nœud jusqu'au destinataire. Chaque message est reçu entièrement pa un nœud avant que celui-ci le retransmette au prochain nœud. Cette technique e utilisée par certaines compagnies aériennes pour la transmission des réservations.

- la technique de *commutation par paquets* est similaire à la commutation de messag à la différence que les messages sont découpés au départ en paquets de taille limité Les paquets sont acheminés jusqu'au destinataire et recomposés ensuite. On gagn ainsi de la souplesse et on améliore les temps de transmission, en particulier parce qu les paquets de faible taille transitent par la mémoire centrale des nœuds et n'ont pa à être conservés dans les disques. Cette technique est utilisée pour le réseau Interne (voir chapitre 3) et par TRANSPAC.

  Les réseaux qui utilisent cette dernière technique peuvent être de deux types :

  - *sans établissement d'un circuit virtuel* : les paquets, appelés *datagrammes*, son transmis totalement indépendamment les uns des autres, par des chemins qu peuvent être différents. Les paquets peuvent donc arriver dans le désordre che le destinataire. C'est à celui-ci de les remettre dans l'ordre. Les vérifications d la bonne arrivée des messages complets ne peuvent s'effectuer que de bout e bout par la machine à qui le message est destiné, et pas à chaque passage pa un nœud. On verra que c'est le service offert par Internet avec le protocole I (voir 3.3). Le protocole TCP (voir 3.4) se chargeant (entre autre) de remettr les paquets dans l'ordre et de vérifier leur bonne arrivée.
  - *avec un circuit virtuel* : un circuit virtuel est établi au début de la connexio entre l'expéditeur et le destinataire du message. Il garantit l'arrivée des paquet dans l'ordre. Tous les paquets suivent le même chemin et le réseau contrôle

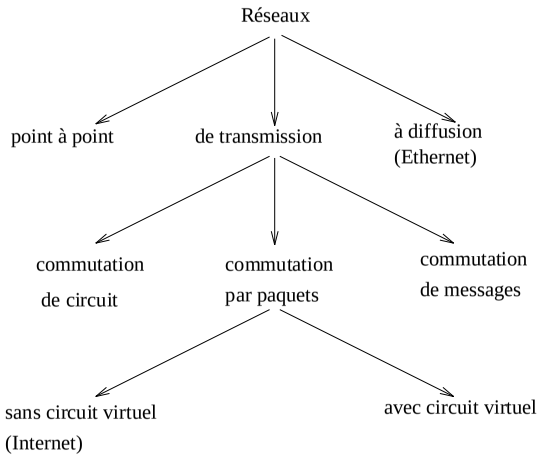

FIG. 1.1 – *Modalités de transmission des informations dans les réseaux*

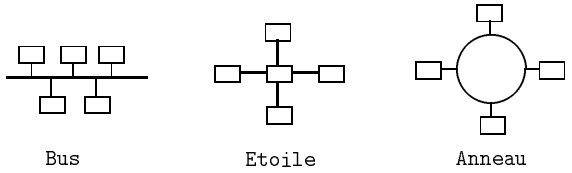

FIG. 1.2 – *Topologies de réseaux locaux*

bonne arrivée des paquets à chaque nœud de passage. C'est le mode de fonctionnement des réseaux de type X.25 comme TRANSPAC.

1.1.3 Topologie

Les réseaux se différencient par leur *topologie* c'est-à-dire par la figure géométrique constituée par les lignes entre les stations. Les réseaux locaux ont le plus souvent des topologies en bus, en étoile ou en anneau (voir figure 1.1). Les réseaux étendus ont des topologies plus complexes.

1.1.4 Protocole

Les réseaux se différencient aussi par les *protocoles* qu'ils utilisent. Pour échanger d
données, les ordinateurs d'un même réseau doivent respecter les mêmes protocoles. E
particulier les données échangées dans le réseau doivent être envoyées dans un format bie
défini.

Ces protocoles, indépendants des machines et des systèmes d'exploitation, permette
l'échange de données dans un réseau hétérogène reliant des types de machines différen
fonctionnant avec des systèmes d'exploitation divers. Ils permettre à deux ordinateurs
parler un langage commun pour s'échanger des informations.

1.2 Le modèle OSI pour les réseaux

1.2.1 Architecture de réseau en couche

Les logiciels réseaux sont souvent complexes de par les multiples domaines qu'ils ont
prendre en compte. La décomposition en couches permet d'obtenir des logiciels plus facil
à écrire et à maintenir. On peut ainsi modifier le type d'accès à un réseau sans remett
en cause les couches hautes des applications existantes.

Une architecture réseau en couches doit respecter les principes suivants:

- Chaque couche offre à la couche immédiatement supérieure les services définis par
  modèle OSI. Elle lui épargne ainsi les détails de la mise en œuvre de ces services.
- Chaque couche ne peut utiliser que les services offerts par la couche immédiateme
  inférieure pour envoyer ou recevoir des messages.
- Chaque couche ne peut échanger (au sens logique) des messages qu'avec une aut
  couche de même niveau (fonctionnant sur un autre ordinateur). Les protocoles d
  finissent la forme des messages échangés entre les couches de même niveau (vo
  figure 1.3 page 7).

1.2.2 Couches du modèle OSI

L'ISO (*International Standard Organisation*) a édicté un standard, le modèle OSI (*Op
System Interconnection*), qui définit 7 couches pour décrire les différentes fonctionnalit
offertes par les réseaux. Ces 7 couches sont résumées dans la table 1.1.

Tous les réseaux ont une architecture en couches mais ils ne respectent pas toujours
modèle OSI. En particulier, le modèle SNA d'IBM ou le modèle TCP/IP qui est deve
un standard de fait dans la communauté Unix, ne suivent pas ce modèle ; les différent
couches ne correspondent pas exactement aux couches du modèle OSI. On étudiera pl
en détail le modèle TCP/IP dans les chapitres 3 et 4.

Cependant le découpage en couches du modèle OSI sert souvent de base à la descriptio
des différents réseaux, même pour les réseaux qui ne respectent pas exactement ce modè

	Nom de la couche	Service rendu	Exemples
7	APPLICATION	Rend les services demandés par l'utilisateur	mail, rlogin, ftp, News, X Window
6	PRESENTATION	Cache les différences dans les structures internes des formats de données, par une mise en forme abstraite des données.	Format des données XDR
5	SESSION	Etablit une session sur une machine distante Offre des possibilités de reprise après incident	Librairie RPC, TCP
4	TRANSPORT	Gère la bonne arrivée des données sans erreur au destinataire final Optimisation de la gestion des ressources réseau	TCP, UDP
3	RESEAU	Routage des paquets Interconnexion des réseaux hétérogènes	X.21, X.25, IP
2	LIAISON	Transmission sans erreur des données entre deux stations directement connectées Découpage des données en trames Contrôle de flux (si émetteur trop rapide)	Ethernet, Token Ring, X.21, X.25, FDDI
1	PHYSIQUE	Transmission des bits sur un support physique	Ethernet, Token Ring, X.21, X.25, FDDI

TAB. 1.1 – *Modèle OSI*

Il faut remarquer que le modèle OSI ne définit ni les protocoles d'échange des données entre chacune des couches de même niveau ni les interfaces entre les différentes couches. Ces protocoles et ces interfaces sont conçus par des travaux ultérieurs de standardisation effectués dans des organismes comme l'ISO par la communauté scientifique et économique.

Les couches "basses" 1 à 3 concernent le réseau, les couches "hautes" 5 à 7 sont plutôt liées aux applications. La couche 4 (couche transport) fait le lien entre ces deux ensembles de couches. C'est la première couche qui assure des liaisons "de bout en bout" : la couche transport de la machine qui envoie le message se met en contact directement avec la couche transport de la machine à qui le message est destiné. Les couches inférieures concernent des liaisons entre machines voisines sans passage par des machines intermédiaires. Par exemple, chaque couche réseau de toutes les machines traversées doit lire le message pour le rediriger (routage) vers une autre machine, en direction de la machine à qui le message est destiné.

1.2.3 Chemin physique et chemin logique

On peut parler du *chemin logique* des données : chaque couche de la machine A dialogu avec la couche de même niveau de la machine B en utilisant les protocoles de ce niveau sans s'occuper des autres niveaux (voir chemins en pointillés sur la figure 1.3 page 7).

Le *chemin physique* réellement suivi par les données est différent. Sur la machine A l données descendent en passant par toutes les couches inférieures. Chaque couche ajout éventuellement des informations imposées par les protocoles de ce niveau pour se fai comprendre de la couche de même niveau de la machine B. Elle peut aussi être amen à effectuer d'autres types de traitement, par exemple à découper les messages en tram plus petites si le protocole l'impose. Les données passent ensuite par le media physique e remontent dans la machine B en passant par toutes les couches de niveau inférieur. A passage ces couches débarrassent les messages des informations ajoutées par la couche d même niveau de la machine A, avant de transmettre les données à la couche de nivea supérieur. Elles peuvent aussi regrouper des trames provenant d'un message découpé pa la couche de même niveau de la machine A.

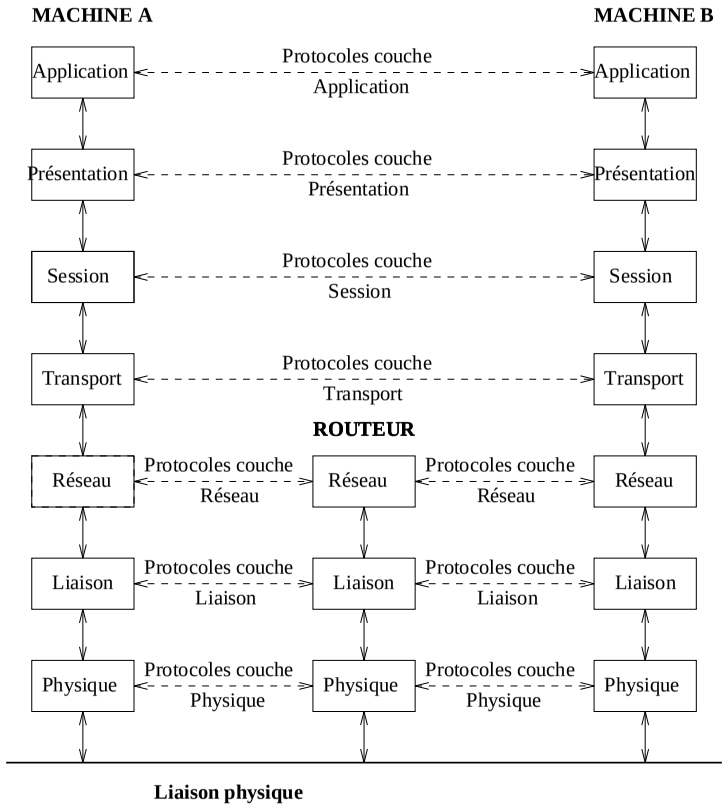

FIG. 1.3 – *Protocoles OSI et cheminement physique des données*

Chapitre 2

Réseaux Ethernet

Ethernet est le type de réseau le plus utilisé actuellement pour les réseaux locaux c machines Unix. Les réseaux locaux de l'Université de Nice sont des réseaux Ethernet.

Ethernet était à l'origine un produit de Xerox mais il est maintenant passé dans domaine public. Il respecte la norme pour réseaux locaux IEEE 802.3 de l'ISO. La norm Ethernet correspond aux couches physique et liaison du modèle OSI.

Un réseau Ethernet a une topologie en bus.

Le câble coaxial fin ou épais était le support physique standard pour le transport d données. Le câble de type paires torsadées (norme 10baseT) est de plus en plus utilisé.

Ethernet est de la famille des réseaux locaux CSMA/CD (*Carrier Sense Multiple A cess/Collision Detection*). Avant d'émettre, une station écoute le réseau et n'émet que elle ne détecte pas d'autres émissions. Le message est envoyé sur tout le réseau; le résea est de type diffusion(voir 1.1.2). Le message contient l'adresse de la station destinatri et un des rôles de l'interface Ethernet d'une station est de filtrer les messages et de r récupérer que ceux qui sont destinés à sa station.

Malgré les précautions prises au moment de l'envoi des messages, deux stations peuve émettre en même temps car la vitesse sur la ligne n'est pas infinie. En cas d'interféren dans les émissions de deux stations, celles-ci détectent la collision et émettent à nouvea les mêmes données à des moments différents.

Ethernet définit le format de la *trame* physique envoyée (unité de communication q contient les informations échangées) : la trame contient un préambule de synchronisatio (servant à synchroniser le récepteur, avant la lecture du restant de la trame), les adresse du destinataire et de l'expéditeur, les données à transmettre et une zone de contrôle pou repérer certaines erreurs de transmission.

Chaque ordinateur connecté à un réseau Ethernet est repéré par une adresse Etherne unique (pour toutes les adresses Ethernet du monde entier) de 48 bits. Cette adresse e liée à la carte Ethernet qui est chargée de l'interface entre l'ordinateur et le réseau.

Le bit le plus significatif de l'adresse de destination d'un message est mis à 0 pou les adresses "individuelles" et il est mis à 1 pour les adresses de groupes (correspondant plusieurs stations). Si tous les bits sont à 1, le message est destiné à toutes les stations.

La vitesse de transfert est de 10 Mbits (mégabits) par seconde. Une norme à 100 Mb

est en cours d'étude. Plusieurs propositions sont en concurrence.

La portée maximale entre deux ordinateurs est de quelques centaines de mètres pour un seul tronçon. En utilisant des *ponts (bridge)* ou des *répéteurs* on peut obtenir des réseaux de quelques kilomètres. Un répéteur ne fait que transmettre les données d'un tronçon à l'autre alors qu'un pont examine les trames pour décider si elles doivent être transmises ou non (c'est inutile si le destinataire de la trame n'appartient pas au tronçon contrôlé par le pont).

Chapitre 3

Réseau Internet, TCP/IP

3.1 Quelques généralités sur le réseau Internet

Les réseaux locaux se sont beaucoup développés dans les 15 dernières années. Très tô
les utilisateurs des réseaux locaux ont ressenti le besoin d'échanger des données entre l
divers réseaux. Ces réseaux étaient souvent incompatibles entre eux avec des matériel
protocoles et logiciels différents.

Des études ont été entreprises pour interconnecter ces réseaux. Il a été nécessaire d'ut
liser des protocoles au dessus des protocoles particuliers de chacun des réseaux.

Le premier grand réseau interconnectant d'autres réseaux a été le réseau Internet (
"*Inter*" et "*Net*work" qui veut dire "réseau" en anglais). Les études et expériences ont ét
développées à l'origine par le DARPA (*Defense Advanced Projects Research Agency*) au
Etats-Unis. Le DARPA a commencé à sponsoriser le réseau ARPANET vers la fin d
années 60. Les protocoles TCP/IP (*Transmission Control Protocol/Internet Protocol*) o
été conçus vers la fin des années 70 et c'est la "*National Science Fondation*" (NSF) qui
décidé de créer le réseau Internet au milieu des années 80 en partant de l'expérience acquis
avec ARPANET.

Le réseau Internet est un réseau mondial qui réunit des réseaux de tous les continents. D
nombreux réseaux dans le monde ont obtenu des points de connexion au réseau Interne
C'est le cas par exemple du réseau universitaire français RENATER (voir 7.3.1) auqu
appartient l'Université de Nice Sophia-Antipolis.

Le réseau Internet a une croissance très rapide. Il double de taille à peu près tous l
ans : au début 1994, 20.000 réseaux et 2.500.000 machines étaient connectés depuis 62 pay
différents. On estime à plusieurs dizaines de millions le nombre d'utilisateurs d'Internet.

Les principaux protocoles utilisés dans Internet sont IP, TCP et UDP (*User Da
Protocol*). De nombreux autres protocoles correspondent aux différents services rendu
par Internet : par exemple ICMP pour les messages de contrôle liés à IP, SNMP pou
l'administration des réseaux ou FTP pour la copie de fichiers.

Internet est géré actuellement par une administration centrale située aux Etats-Un
(le NIC : *Network Information Center*) à qui toute demande de nouvelle connexion d'u

réseau doit être adressée.

Les réseaux sont hiérarchisés en domaines et sous-domaines. Chaque pays correspond à un domaine de premier niveau ("*top-level domain*") qui est géré localement. En France la gestion est confiée à l'INRIA.

3.2 Principes généraux de fonctionnement d'Internet

Internet a choisi la technique de commutation de paquets sans établissement d'un circuit virtuel (voir 1.1.2).

Le deuxième principe choisi par Internet est que l'interface entre deux réseaux utilisant des protocoles de transmission différents ne se fera pas par traduction des protocoles mais par ajout dans les messages des informations liées à un protocole commun aux deux réseaux (protocole IP).

L'interface entre les différents réseaux est constituée par des *routeurs* : des ordinateurs qui appartiennent à plusieurs réseaux à la fois et qui sont chargés de transmettre les paquets de données d'un réseau à l'autre. Des tables de routage indiquent aux routeurs les chemins à suivre pour atteindre les différents réseaux connectés à Internet. Ces tables sont conservées dans des fichiers système de chaque routeur.

Lorsqu'un message envoyé par un ordinateur est destiné à un autre réseau d'Internet, le message passe par plusieurs routeurs jusqu'à atteindre le réseau local qui contient l'ordinateur destinataire.

Le dernier routeur doit alors trouver à quel ordinateur de son réseau le message est destiné. Le message contient en effet l'adresse Internet (adresse IP plus exactement) qui n'est pas l'adresse physique (adresse Ethernet par exemple). La traduction d'une adresse IP en adresse physique se fait par diffusion : le routeur envoie à tous les ordinateurs du réseau (local) un message demandant quelle est la machine qui a l'adresse IP spécifiée dans le message qu'elle vient de recevoir. La machine qui a cette adresse répond en donnant son adresse physique. Le routeur peut alors envoyer le message. La réponse indiquant l'adresse physique est conservée dans un cache pour que les connexions suivantes se fassent plus rapidement.

3.3 Protocole IP, adresses IP

Grâce à l'essor formidable d'Internet, TCP/IP est devenu une norme de fait pour la connexion des réseaux hétérogènes.

Les services rendus par le protocole IP correspondent à peu près à la couche réseau du modèle OSI. Il définit en particulier le format des paquets de données échangés : les *datagrammes* IP.

Chaque ordinateur relié au réseau est repéré par une adresse IP de 32 bits, composée d'une partie liée à la machine elle-même et d'une partie qui est liée au réseau auquel

appartient la machine[1].

L'adresse des réseaux est déterminée par le NIC (voir 3.1) qui distribue des adresses d
3 types pour les réseaux de classes A, B ou C :

classe A : le premier bit est *0*, les 7 bits suivants désignent le réseau et les 24 bits restant
désignent la machine sur le réseau. Cette classe correspond aux très grands réseau
comportant un grand nombre de machines. On a donc 1 octet pour identifier le résea
et 3 pour identifier une machine du réseau.

classe B : les 2 premiers bits sont *10*, les 14 bits suivants indiquent le réseau et les 16 bi
restants désignent la machine sur le réseau (2 octets pour le réseau et 2 pour un
machine).

classe C : les 3 premiers bits sont *110*[2], les 21 bits suivants désignent le réseau et les
bits restants désignent la machine sur le réseau (3 octets pour le réseau et 1 pour un
machine). La classe C correspond aux petits réseaux.

Il est possible de découper un réseau en sous-réseaux (un seul niveau de décomposition
une partie des bits réservés à la machine peut désigner un *sous-réseau*. Le nombre de bi
est défini par un masque binaire de 32 bits : tous les bits d'une adresse qui correspondent
à un bit mis à 1 dans le masque désignent le sous-réseau.

Les adresses IP sont ordinairement écrites par groupes de 8 bits donnés sous form
décimale, séparés par des points. On remarquera que le premier nombre de l'adresse I
d'un ordinateur indique le type du réseau auquel il appartient : nombre inférieur à 128 pou
les réseaux de classe A, compris (au sens large) entre 128 et 191 pour les réseaux de class
B et compris (au sens large) entre 192 et 223 pour les réseaux de classe C.

Par exemple la machine taloa du réseau de l'université de Nice (qui est de classe B),
pour adresse 134.59.0.7. 134.59 désigne le réseau et 0.7 la machine sur le réseau.

Il n'est pas pratique pour l'utilisateur d'utiliser des adresses IP. Le service DNS (*Do
main Name System*) permet d'utiliser des noms de domaines pour désigner les machines d
réseau. Les domaines sont hiérarchisés en arbre. Le premier niveau correspond à des pay
("fr" pour France). Quelques noms de trois lettres correspondent à des types d'utilisateu
("edu" pour le monde de l'éducation aux États-Unis, "com" pour les entreprises comme
ciales). Ces domaines de haut niveau ont des sous-domaines qui peuvent eux-mêmes êtr
subdivisés en sous-domaines.

La correspondance entre ces noms et les adresses IP est effectuée par des ordinateu
serveur de noms. Les ordinateurs du Campus Valrose forment le domaine "unice.fr". L
serveur de nom local est la machine taloa, de nom complet "taloa.unice.fr".

Il faut remarquer qu'il n'y a pas correspondance entre les adresses IP et la hiérarch
des noms de domaine. Par exemple, toutes les adresses des machines de l'INRIA ont u
nom du type "machine.inria.fr", qu'elles soient de l'INRIA Rocquencourt près de Paris o

1. L'explosion d'Internet a provoqué une pénurie d'adresses IP disponibles. Dans la nouvelle version
IP (IP v6) les adresses seront codées sur 16 octets

2. l'adresse composée uniquement de 1 est une pseudo-adresse désignant toutes les machines du rése

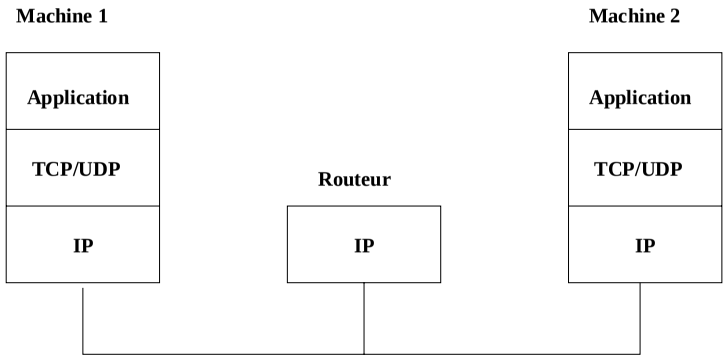

FIG. 3.1 – *Architecture TCP/IP*

de l'INRIA Sophia Antipolis. Pourtant le réseau de Rocquencourt n'a pas du tout le même numéro de réseau que celui de Sophia Antipolis.

Les adresses et les noms des machines du réseau Internet visibles d'une machine sont enregistrés dans le fichier `/etc/hosts` de la machine ou dans le système NIS s'il est installé (voir 6.1).

Les stations émettrices et destinatrices peuvent se trouver dans deux réseaux différents. Un système de routage permet de déterminer un chemin entre les deux stations en passant par des routeurs (voir 3.2). Certains auteurs utilisent le terme de "passerelle" ("*application gateway*" en anglais) au lieu de routeur, mais il est d'usage de le réserver à des machines qui font la conversion entre protocoles différents au niveau application du modèle OSI (par exemple, pour convertir des messages entre deux formats différents de courrier électronique).

IP envoie des datagrammes d'un endroit à l'autre du réseau mais il n'assure pas le contrôle de la bonne arrivée des données (possibilité de perte, duplication, arrivée dans un mauvais ordre des paquets).

3.4 Protocole TCP

TCP est placé au-dessus de IP ; il correspond à peu près à la couche transport et à une partie de la couche session du modèle OSI.

TCP assure une communication sûre en ouvrant une connexion "*full-duplex*" (possibilité pour les deux machines d'émettre en même temps) entre deux machines et en vérifiant qu'il n'y a pas d'erreur dans la transmission des données.

TCP assure un contrôle de bout en bout : si la liaison entre l'émetteur et le destinataire passe par plusieurs routeurs, le contrôle se fait à chaque bout de la connexion et pas sur chaque liaison entre routeurs. La figure 3.1 montre la connexion de deux machines situées

sur deux réseaux différents. On voit que TCP n'intervient pas sur le routeur, mais seulemen aux deux extrémités de la connexion.

TCP divise les données en segments qui sont insérés dans des datagrammes IP. A chaqu réception d'un segment, un accusé de réception est renvoyé à l'expéditeur. Si l'expéditeu ne reçoit pas cet accusé de réception avant un certain temps, il réexpédie le datagramme

Chaque application TCP (*rlogin, telnet, rsh,...* étudiées au chapitre 4) correspond un numéro (de 16 bits) de *port* normalisé qui permet de savoir à quelle application segment est destiné. On trouve ces numéros dans le fichier `/etc/services`. L'adresse d l'ordinateur destinataire est indiquée dans le datagramme IP. Le port qui indique à quel application le message est destiné est enregistré dans la partie du message spécifique TCP. Ces numéros de ports ne changent jamais (ou très rarement) et ils sont gérés par un administration centrale.

3.5 Protocole UDP

UDP (User Data Protocol) correspond à TCP mais il ne gère pas la bonne arrivée de données (ce n'est pas un protocole complet pour la couche "transport").

C'est un protocole sans connexion qui se contente d'envoyer des datagrammes en ajou tant seulement au dessus de IP la possibilité de :

- détecter des erreurs à l'arrivée par l'ajout d'un "*checksum*" aux données transmise
- distinguer entre les diverses applications travaillant dans un même ordinateur pa l'indication d'un port comme pour TCP. Comme pour TCP, on trouve ces numéro dans le fichier `/etc/services`.

UDP est souvent utilisé pour construire un protocole plus complet. NFS (voir 6.1) e construit actuellement au dessus de UDP.

3.6 Protocole ICMP

ICMP (Internet Control Message Protocol) est un protocole requis par IP pour l'env de messages de contrôle.

La plupart des messages de contrôle informent l'émetteur d'un paquet que celui-ci a ét détruit par un routeur par exemple parce que la machine destinatrice n'était pas accessib ou parce que le paquet avait séjourné trop longtemps sur le réseau (le plus souvent à caus d'une boucle dans les circuits de routage).

ICMP offre aussi la possibilité d'envoyer des messages d'"écho". Lorsqu'ils sont reçu ces messages engendrent automatiquement un message envoyés à l'émetteur du messag d'écho. Ces messages sont utilisés par la commande *ping* (voir 4.9.2).

3.7 Activité réseau sur une machine (netstat)

La commande *netstat* affiche l'activité réseau de la machine sur laquelle on s'est connecté. Pour plus de détails, taper "`man netstat`".

Chapitre 4

Commandes au-dessus de TCP/IP

4.1 Généralités

Dans ce chapitre nous allons décrire des commandes Unix qui utilisent TCP/IP pou fournir des services à l'utilisateur via une connexion à une machine distante. Ces servic offerts (en particulier la connexion à distance sur une autre machine et la copie de fichie distants) ainsi que les services décrits au chapitre 5 ont fait le succès du réseau Interne dans le monde universitaire.

Pour la clarté des exposés à venir, on appellera *machineL* la machine locale sur laquel on est déjà connecté, *machineD* la machine distante sur laquelle on veut se connecter ; *nom* le nom d'utilisateur sous lequel on est entré sur la machineL et *nomD* le nom d'utilisateu sous lequel on veut entrer sur la machineD.

4.2 Principes de fonctionnement des applications TCP/

Ces commandes correspondent à la couche "application" du modèle OSI.

Elles fonctionnent sur le modèle *client-serveur*: un programme "serveur" rend des se vices à un programme "client". Le programme client exprime ses besoins par des requêtes qu vont transiter sur le réseau pour atteindre le programme serveur. Le programme serveur e un "daemon" (acronyme de *deferred auxiliary execution monitor*, c'est à dire, programm de contrôle d'exécution différée) qui fonctionne en arrière-plan pour répondre aux requête du client adressées au port correspondant à l'application (voir 3.4). Un protocole est assoc à chaque type d'application TCP/IP. Ce protocole détermine en particulier le format de requêtes du client et des réponses du serveur.

Chaque application TCP/IP (telnet, ftp, etc.) qui tourne sur une machine a son propr serveur mais, en pratique, pour éviter qu'il y ait trop de processus en exécution sur un machine, un seul daemon (*/etc/inetd*) scrute les différents ports liés aux services spécifié dans le fichier `/etc/inetd.conf`. A l'arrivée d'une requête, ce daemon réveille le daemo concerné.

4.3 Format des adresses Internet

Dans les commandes qui seront étudiées par la suite, l'adresse d'un machine sera son adresse IP ou le nom Internet de la machine, l'adresse d'un utilisateur sera du format suivant : *nom-login*@*nom-machine*

Exemple 4.1

```
toto@taloa.unice.fr
```

L'association d'un nom à une machine est fourni par le service "*Domain Name System*" (DNS) (voir 3.3) construit au dessus de TCP/IP.

Le fichier `/etc/hosts` contient l'adresse et le nom Internet des machines connues (de la machine locale) que l'on peut joindre par le réseau Internet. Chaque machine est décrite par une ligne de ce fichier.

Remarque 4.1

Si le système NIS fonctionne (pour partager les données entre plusieurs machines), les commandes qui permettent d'avoir les informations sur les machines sont *ypcat* et *ypmatch* (voir cours sur Unix).

4.4 Condition nécessaire pour les autorisations d'accès

Nous allons exposer d'abord une condition nécessaire pour que la connexion sur *machineD* avec le nom de login *nomD* soit possible :

nomD doit être enregistré comme un nom d'utilisateur sur *machineD*

On ne peut donc rien faire sur une machine si on ne dispose pas d'un compte sur cette machine. On verra que, lorsque l'on travaille avec ftp, on peut disposer d'un compte spécial nommé "*anonymous*" qui permet de copier des fichiers d'une machine même si l'on ne dispose pas réellement d'un compte sur cette machine.

D'autres conditions pourront se rajouter à ces conditions pour chaque commande particulière.

4.5 Accorder sa confiance à des utilisateurs sur d'autres machines

4.5.1 Groupes du réseau

Le système NIS offre la possibilité de définir des groupes formés de machines, d'utilisateurs ou d'utilisateurs sur des machines particulières.

`ypcat -k network`
affiche ces groupes. Si vous voulez des précisions sur le format affiché, tapez "`man network`".

Ces groupes peuvent être utilisés dans les fichiers "rhosts" étudiés dans la section su
vante.

Exemple 4.2

```
BD (lucky.unice.fr,-,mipsnis)  (achille.unice.fr,-,mipsnis)
PROFS (-,jpq,mipsnis) (-,jm,mipsnis) (-,mft,mipsnis) PROF2
```

définit un groupe "BD" de machines et un groupe "PROFS" d'utilisateurs.

4.5.2 Fichiers "rhosts"

Pour les commandes que nous étudierons ici, les deux fichiers `/etc/hosts.equiv` e
`.rhosts` du répertoire HOME de l'utilisateur *nomD* jouent un rôle particulier :

Ces deux fichiers de *machineD* (la machine sur laquelle on veut se connecter) peuver
contenir des lignes du type

machineL

qui indique que *machineD* peut avoir "confiance" dans tout utilisateur de *machineL* q
voudra se connecter à *machineD sous le même nom* que son nom de login sur *machineL*

Dans le fichier `.rhosts`, l'utilisateur *nomD* de *machineD* peut aussi indiquer qu'il
confiance en un *utilisateur nomL* de *machineL* et qu'il l'autorise à se connecter sous so
nom *nomD* sur *machineD* :

machineL nomL

Le fichier `/etc/hosts.equiv` est écrit par l'administrateur du système. Le simple ut
lisateur ne peut donc jouer que sur son fichier `~/.rhosts`. Le plus souvent il se désigr
lui-même sur une autre machine (machineL) pour pouvoir se connecter facilement sur *ma*
chineD à partir de *machineL*.

Remarque 4.2

Les facilités offertes par ces fichiers nuisent à la sécurité et il faut les utiliser ave
prudence. Il est préférable d'avoir un mot de passe différent sur chacune des machine
que l'on utilise pour mieux se protéger si un de ses mots de passe est découvert. S
on utilise des fichiers `.rhosts`, la découverte d'un des mots de passe ouvre toutes l
autres machines.

Dans ces fichiers on peut utiliser des groupes du réseau (voir section précédente) :

+@*groupe*

donne sa confiance à tous les membres du *groupe*. Si le groupe désigne des machines, o
peut ajouter un nom d'utilisateur à la suite du nom du groupe pour limiter sa confiance
cet utilisateur sur ces machines.

Exemple 4.3

Si un fichier "`.rhosts`" du répertoire "HOME" de l'utilisateur "toto" contient

```
samoa.unice.fr
clio.unice.fr rtoto
+@BD
```

l'utilisateur "toto" pourra se connecter sans donner de mot de passe de la machine "samoa.unice.fr, des machines du groupe "BD" et l'utilisateur "rtoto" pourra faire de même depuis la machine "clio.unice.fr".

4.6 Connexions à distance

4.6.1 Entre 2 systèmes Unix compatibles (rlogin)

La commande *rlogin* permet de se connecter sur un ordinateur distant fonctionnant sous un Unix compatible avec le système Unix de la machine locale.

`rlogin` *machineD* `[-l` *nomD*`]`

Si l'utilisateur est entré sur un ordinateur *machineL* connecté à un réseau, il peut se connecter à une autre machine *machineD* du réseau depuis cet ordinateur. Il peut se connecter sous le même nom que son nom de login (*nomL*) sur le premier ordinateur mais il peut aussi entrer sous un autre nom *nomD* (option -l).

Le shell de login de l'utilisateur *nomD* est lancé sur *machineD*. Si le shell est ksh, `.profile` et le fichier dont le nom est contenu dans la variable ENV de l'environnement distant (`.kshrc` le plus souvent) sont exécutés. Si le shell est zsh, `.zshenv`, `.zlogin` et `.zshrc` sont exécutés.

Le contrôle des sorties sur la sortie standard par [Ctrl] Q et [Ctrl] S fonctionne. La variable TERM de l'environnement local de machineL est transmise à l'environnement de machineD. Les autres variables ne sont pas transmises.

On peut suspendre provisoirement la connexion en tapant "`~[Ctrl]Z`" en début de ligne. On la reprend ensuite en tapant la commande `fg` (sous ksh ou zsh).

Autorisations d'accès:

La condition nécessaire de connexion définie en 4.4 doit évidemment être satisfaite : *nomD* doit être enregistré comme utilisateur sur *machineD*.

De plus, on n'aura pas à entrer le mot de passe de *nomD* dans l'un des deux cas suivants :

- si *machineL* est enregistrée dans le fichier `/etc/hosts.equiv` de *machineD*, si on veut se connecter à *machineD* et que l'on n'a pas utilisé l'option `-l`,
- si *machineL* est enregistrée dans le fichier `.rhosts` du répertoire HOME de l'utilisateur *nomD* sur *machineD* et que l'on n'a pas utilisé l'option "`-l`",
- sinon, si *nomL* sur *machineL* est enregistrée dans le fichier `.rhosts` du répertoire HOME de l'utilisateur *nomD* sur *machineD*

Exemples 4.4

(a) `rlogin mimosa`

(b) `rlogin mpim-bonn.mpg.de -l toto`

Si l'utilisateur titi de la machine taloa.unice.fr a tapé cette commande, il pourra entrer comme utilisateur toto sur la machine *mpim-bonn.mpg.de* sans taper de mot de passe si le fichier .rhosts de l'utilisateur toto de la machine *mpim-bonn.mpg.d* contient la ligne suivante :

`taloa.unice.fr titi`

4.6.2 Entre 2 systèmes d'exploitation différents (telnet)

Si on veut se connecter à un ordinateur qui ne travaille pas avec le système d'exploit tion Unix, il faut utiliser la commande *telnet*.

`telnet [`*machineD*`]`

Si on précise le nom de la machine distante *machineD*, on se retrouve en début d session sur cette machine avec le plus souvent un nom de login et un mot de passe à entr (les fichiers du type ".rhosts" ne sont pas utilisés par telnet). On récupère la main sur machine locale dès que la session se termine sur la machine distante.

Si on ne précise pas la machine, on entre dans le mode commande de telnet (avec prompt "telnet>").

Dans ce mode, la commande "quit" permet de sortir de telnet. La commande "?" affich les commandes disponibles qui permettent de se connecter à d'autres machines. Nous n les détaillerons pas ici.

4.7 Exécution d'une commande Unix distante (rsh)

`rsh` *machineD* `[-l` *nomD*`] [-n] [`*commandeD*`]`

lance un shell sur *machineD* sous le nom d'utilisateur *nomD* (par défaut, sous le mêm nom que le nom de login local *nomL*). Le shell est le shell de démarrage (celui qui est lanc au login) de *nomD*.

Si *commandeD* est omise *rsh* est équivalente à *rlogin*.

Option

`-n` redirige automatiquement l'entrée standard vers `/dev/null` ; voir remarque (ci-dessous

Les fichiers d'initialisation lancés au démarrage du shell sont exécutés :

- si *commandeD* est omise, voir *rlogin* en 4.6.1,
- si *commandeD* est indiquée, il s'agit d'un shell non interactif *et qui n'est pas un she de login*. Aucun fichier n'est exécuté si le shell est ksh (car la variable `ENV` n'a re aucune valeur sur *machineD*) et les fichiers d'initialisation `/etc/zshenv` et `.zsher` sont exécutés si le shell est *zsh*.

rsh redirige ses entrée et erreur standards vers les entrée et erreur standards de *commande* et copie la sortie standard de la *commande* vers sa sortie standard. On peu ainsi utiliser rsh avec des pipes ou des redirections.

Les interruptions sont transmises à la *commande* distante mais pas l'environnemen local (les variables).

Autorisations d'accès:

La condition nécessaire de connexion définie en 4.4 doit évidemment être satisfaite : *nomD* doit être enregistré comme utilisateur sur *machineD*.

Si rsh est lancé sans l'option "-l *nomD*", la ligne

machineL

doit être enregistrée dans le fichier "`/etc/hosts.equiv`" ou dans le fichier "`.rhosts`" du répertoire HOME de l'utilisateur nomD sur machineD.

Si rsh a été lancée avec l'option "-l *nomD*", la ligne

machineL nomL

doit être enregistrée dans le fichier "`.rhosts`" du répertoire HOME de l'utilisateur *nomD* sur *machineD*.

Remarques 4.3

(a) Si le shell de login de *nomD* est ksh, aucun fichier d'initialisation de la machine distante n'est exécuté. Certaines variables utiles à l'exécution de *commandeD* risquent donc de ne pas avoir les valeurs correctes, en particulier la variable `PATH` qui permet de retrouver l'endroit où se trouve *commandeD*. Il est donc prudent de donner le chemin absolu de la *commandeD* distante. On peut aussi initialiser un environnement minimum en préfixant *commandeD* par des affectations de variables (voir cours Unix) :
rsh *machineD* 'var1=val1 var2=val2 *commandeD*'

Si le shell de login de *nomD* est zsh, `.zshenv` est exécuté ce qui, le plus souvent, positionne correctement au moins la variable `PATH`.

(b) Le shell local interprète normalement la ligne de commande avant de lancer rsh. Si on veut envoyer des caractères spéciaux au shell distant (le shell de login de *nomD* sur *machineD*), on doit empêcher leur interprétation par le shell local (par exemple en les entourant d'apostrophes).

(c) Dans certains cas, en particulier si rsh est lancé en arrière-plan, il est nécessaire de rediriger l'entrée standard de rsh vers `/dev/null` sinon le processus qui exécute rsh est bloqué à cause des interactions entre le shell de travail et rsh. En effet les commandes lancées en arrière-plan qui ont le terminal comme entrée standard sont bloquées pour éviter qu'elles ne "volent" les caractères entrés au clavier par l'utilisateur (voir option "`-n`")

Exemples 4.5

(a) `rsh taloa cat fichierD > fichierL`
copie le fichier "`fichierD`" de la machine taloa sur le fichier "`fichierL`" de la machine locale.

(b) `rsh taloa 'cat fichierD1 > fichierD2'`
copie le fichier "`fichierD1`" de la machine taloa sur le fichier "`fichierD2`" de la même machine taloa.

(c) `rsh taloa xappli -display liline:0.0 < /dev/null &`
lance la commande X Window "`xappli`" de la machine taloa pour travailler sur l station de travail "`liline`".

(d) `rsh taloa 'ls ~/f* | cpio -o' > /dev/rmt0h`
sauvegarde les fichiers de son répertoire HOME de taloa, dont le nom commen par la lettre f, sur le streamer (associé au fichier spécial `/dev/rmt0h`) connecté à machine locale d'où l'on a lancé rsh. La commande cpio est utilisée pour effectu des sauvegardes. "cpio -o" recopie sur la sortie standard les fichiers dont les noms l sont donnés sur l'entrée standard.

4.8 Copie de fichiers distants

4.8.1 Entre 2 systèmes Unix compatibles (rcp)

```
rcp [-p] fichier1 fichier2
rcp [-rp] fichiers... répertoire
```

La syntaxe est la même que celle de la commande cp. La deuxième forme permet c copier des arborescences entières.

Par défaut, les autorisations et le propriétaire de *fichier2* sont conservés si *fichier2* exis déjà. Sinon, *fichier2* a les autorisations de *fichier1* modifiées par le "masque" sur *machine* (voir commande *umask* dans le cours Unix).

Les noms de fichiers ont le format suivant (en Unix OSF)[1] :

[[*nomD*@]*machineD*:]*chemin*

où *nomD* est le nom de login sous lequel l'utilisateur sera considéré sur la machine *mach neD*.

Exemples 4.6

(a) `taloa:bin/x`
désigne le fichier `bin/x` du répertoire HOME de l'utilisateur *nomL* sur la machi taloa.

(b) `titi@taloa.unice.fr:bin/x`
désigne le fichier `bin/x` du répertoire HOME de l'utilisateur "titi" sur la machi taloa. Dans ce cas, *nomL* entre sous le nom titi sur taloa.

Les chemins relatifs de fichiers de *machineD* sont relatifs au répertoire HOME de *nom*

Options

-p conserve la date de modification et les autorisations des fichiers copiés *sans* ten compte du "masque" local

-r copie toute l'arborescence des répertoires qui sont dans la liste des *fichiers*

1. une autre syntaxe est utilisée en BSD 4.2 et 4.3 : [*machineD*[.*nomD*]:]*chemin*, par exemple, t loa.unice.fr.titi:bin/x

Autorisations d'accès:
Les mêmes que pour rsh.

Exemple 4.7
`rcp taloa:bin/pgm bin`
copie le fichier bin/pgm du répertoire HOME de l'utilisateur sur taloa, dans son répertoire bin sur la machine locale.

Remarques 4.4
(a) Comme pour rsh, il faut empêcher l'interprétation des caractères spéciaux par le shell local si l'on veut qu'ils soient interprétés par le shell distant.

(b) rcp ne permet pas de copier des fichiers entre deux machines si une des machines ne fonctionne pas sous Unix, ou même si les deux machines fonctionnent sous des Unix incompatibles.

4.8.2 Entre 2 systèmes d'exploitation différents (ftp)

Description de la commande

ftp permet la copie de fichiers entre un système Unix et un système d'exploitation différent.

ftp utilise un protocole spécial FTP pour le transfert de fichiers. Ce protocole permet à un utilisateur de se connecter sur un ordinateur distant, de copier des fichiers d'un système à l'autre et d'effectuer des opérations utiles à ces transferts de fichiers : lister des répertoires, se déplacer dans l'arborescence.

`ftp [-in]` ***machineD***

ouvre une connexion avec machineD, demande le nom de l'utilisateur qui sera utilisé sur la machine distante et attend des commandes de l'utilisateur (le prompt est "`ftp>`").

Si un fichier `.netrc` existe dans le répertoire HOME de l'utilisateur qui a lancé ftp, il est lu et les entrées éventuelles qu'il contient indiqueront le nom d'utilisateur sous lequel ftp se connectera sur *machineD*. Voici un exemple de fichier `.netrc` :

```
machine nimbus.unice.fr login titi
machine ftp.inria.fr login anonymous password toto@taloa.unice.fr
```

La première ligne indique que la commande "`ftp nimbus.unice.fr`" ouvrira une connexion sur `nimbus.unice.fr` sous le nom titi. Un mot de passe devra être entré. La deuxième ligne donne en plus le mot de passe pour un "ftp anonymous" (voir 4.8.2) sur la machine `ftp.inria.fr`. Il est évident que pour des raisons de sécurité il faut éviter d'entrer un mot de passe pour un autre compte qu'un compte "anonymous".

Options

-i supprime les messages de confirmation de transfert affichés par les commandes de copies multiples (mget ou mput)

-n le fichier `.netrc` ne sera pas lu au démarrage de ftp

Voici les principales commandes disponibles sous ftp :

quit quitte ftp

help affiche la liste des commandes disponibles

help ***commande***
affiche une aide sur la commande spécifiée

cd ***répertoireD***
changement de répertoire sur la machine distante

lcd [*répertoireL***]**
"`cd`" sur la machine locale

dir [*répertoireD***]**
"`ls -l`" sur la machine distante

ls [*répertoireD***]**
"`ls`" sur la machine distante

pwd "`pwd`" sur la machine distante

ascii

binary permettent de fixer le type de transfert ("ascii" par défaut). Il faut choisir "binary" pour les transferts de programmes et "ascii" pour les transferts de fichiers qui contiennent du texte, en particulier si le transfert se fait entre deux systèmes différents car ftp prend en compte les différents codes pour marquer les fins de lignes

status affiche des informations sur la connexion en cours, par exemple, le type de transfert (ascii ou binaire)

get ***fichierD* [*fichierL*]**
copie *fichierD* de la machine distante sur *fichierL* de la machine locale (nom terminal de *fichierD* par défaut). "-" désigne la sortie standard locale (l'écran). On peut ainsi afficher un fichier distant sans le transférer. Par exemple, "`get README -`" affiche le contenu du fichier `README`. Ctrl C interrompt le transfert du fichier.

mget ***fichiersD...***
copie plusieurs fichiers de *machineD* sur *machineL* (avec des noms inchangés). On peut utiliser des caractères spéciaux comme * pour désigner les fichiers de la machine distante. Les caractères acceptés dépendent du système installé sur *machineD*. Il est prudent de lister les noms des fichiers avec la commande ls avant de lancer mget. On remarquera que sur les machines Unix, "`mget *`" transfère les fichiers du répertoire courant distant *et* ceux qui sont placés directement sous les répertoires placés dans le répertoire courant (en effet, "`*`" sera remplacé par la liste des fichiers affichés par "`ls *`" et mget sera exécuté sur cette liste)

put ***fichierL* [*fichierD*]**
copie un fichier local sur la machine distante

mput ***fichiersL...***
copie plusieurs fichiers locaux (avec un nom inchangé) sur la machine distante. On peut utiliser les caractères spéciaux *,? et [ ] (mais pas le ! ou le ^ juste après le crochet ouvrant pour indiquer une négation)

prompt

indique à ftp s'il doit demander une confirmation pour la copie de chaque fichier lors des copies multiples (mget ou mput)

! [*commande*]
lance une commande sur la machine locale (shell par défaut). Par exemple, "`! ls bin`"
liste le répertoire bin de la machine locale

open *machineD*
établit une connexion avec *machineD*

close ferme la connexion avec la machine en cours

mkdir *répertoireD*
"`mkdir`" sur la machine distante (un seul répertoire)

rmdir *répertoireD*
"`rmdir`" sur la machine distante (un seul répertoire)

delete *fichierD*
supprime un fichier sur la machine distante

mdelete *fichierDs...*
supprime des fichiers de la machine distante. On peut utiliser les mêmes caractères spéciaux que ls ; par prudence il est préférable de lancer ls avant de lancer mdelete pour avoir la liste exacte des fichiers qui seront suprimés (voir commande mget décrite ci-dessus)

rename *ancien-nomD nouveau-nomD*
renomme un fichier distant

macdef macro permet de définir une macro (ensemble de commandes ftp regroupées sous un nom). Cette possibilité est surtout intéressante pour lancer des transferts automatiques avec une macro nommée "init" (voir exemple à la fin de la section sur les ftp "anonymous").

Remarque 4.5
Si le premier caractère d'un nom de fichier est le caractère "|", la sortie standard est redirigée vers la commande locale spécifiée à la suite du "|". Par exemple, "`dir | more`" affiche un listing page à page. Si la commande locale comporte plusieurs mots, elle doit être entre guillemets.

Remarque 4.6
La plupart des versions actuelles de ftp ne peuvent copier que des fichiers et pas des arborescences entières comme la commande rcp. Cependant certains serveurs ftp offrent cette possibilité, en particulier les serveurs "CTAN" (*Comprehensive TEX Archive Network*) qui distribuent les logiciels utilisés par LaTeX en "ftp anonymous". Il suffit d'ajouter au nom du répertoire le suffixe "`.tar`" pour récupérer toute l'arborescence du répertoire sous forme d'un fichier "`.tar`" (voir 4.8.4). On peut aussi recevoir des versions compactées des fichiers ou des répertoires en faisant suivre leur nom des

suffixes correspondants.

Exemples 4.8
(a) `get rep.tar`
récupère toute l'arborescence du répertoire "rep" sous forme d'un fichier "rep.ta
(b) `get rep.tar.Z`
récupère toute l'arborescence du répertoire "rep" sous forme d'un fichier con pacté "rep.tar.Z" (voir 4.8.4).

De plus, ces serveurs permettent de rechercher des fichiers sur le serveur. On tap la commande suivante où *chaîne* est une expression régulière (voir cours Unix) q désigne les noms des fichiers que l'on recherche :

`quote site index` *chaîne*

ftp anonymous

ftp est très utilisé pour copier des logiciels du domaine public par "*anonymous ftp*" certaines machines ont un pseudo-utilisateur de nom de login "anonymous". On peut entre sous ce nom dans le système pour copier des logiciels ou des fichiers. Comme mot de pass on peut entrer ce que l'on veut mais il est d'usage d'entrer son adresse électronique ; cet information peut être utilisée par l'administrateur de la machine serveur distante.

Exemple 4.9

```
$ ftp machine.univ-mrs.fr
Connected to lmd2.univ-mrs.fr.
220 lmd2 FTP server (SunOS4.1) ready.
Name (lmd2.univ-mrs.fr:grin): toto
331 Password required for toto.
Password: XXXXXXXXX
230 User toto logged in.
ftp> cd outils
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (134.59.0.7,4997) (0 bytes).
Conseil.tex
Lettres.tex
T2.tex
226 ASCII Transfer complete.
173 bytes received in 0.016 seconds (11 Kbytes/s)
ftp> prompt
```

```
Interactive mode off.
ftp> mget *
200 PORT command successful.
150 ASCII data connection for Conseil.tex (134.59.0.7,1027) (1529 bytes).
226 ASCII Transfer complete.
local: Conseil.tex remote: Conseil.tex
1602 bytes received in 0.16 seconds (10 Kbytes/s)
200 PORT command successful.
150 ASCII data connection for Lettres.tex (134.59.0.7,1028) (1497 bytes).
226 ASCII Transfer complete.
local: Lettres.tex remote: Lettres.tex
1573 bytes received in 0.14 seconds (11 Kbytes/s)
200 PORT command successful.
150 ASCII data connection for T2.tex (134.59.0.7,1029) (2263 bytes).
226 ASCII Transfer complete.
local: T2.tex remote: T2.tex 2365 bytes received in 0.18 seconds
(13 Kbytes/s)
ftp> quit
221 Goodbye.
```

Dans les différents serveurs "ftp anonymous" on peut trouver tous les logiciels du domaine public (distribués gratuitement) fonctionnant sur les machines les plus utilisées : Unix, PC, Macintosh, Atari, etc. La difficulté quand on cherche un logiciel ou un fichier est de savoir sur quel serveur il se trouve. On dispose pour cela d'outils (étudiés en 5.3). Au fil des recherches on se constitue aussi sa propre liste de serveurs préférés. Les serveurs sont le plus souvent généralistes mais quelques-uns sont spécialisés dans un ou plusieurs domaines.

Voici pour commencer, quelques adresses de serveurs ftp (n'utilisez les serveurs étrangers que si vous n'avez pas trouvé ce que vous cherchiez sur les serveurs français) :

ftp.inria.fr	serveur généraliste
ftp.gnu.ai.mit.edu	pour les logiciels "GNU"[2]
ftp.loria.fr	serveur "CTAN" pour le monde LaTeX

Internet étant souvent surchargé, il est intéressant de savoir faire des transferts automatiques aux heures où il y a moins de charge (par exemple, vers 4 heures du matin pour les transferts des États-Unis). Voici, sans entrer dans les détails, un exemple de procédure pour lancer un transfert automatique même lorsque l'on n'est pas connecté :

1. Écriture d'un shellscript (nommé ici `~/reptransftp/ftp.at`) :

```
cd /users/home/toto
ftp ftp.loria.fr
```

2. GNU est un projet de la "*Free Software Foundation*" (FSF) pour fournir un environnement de travail gratuit aux utilisateurs

2. Ajout d'une entrée dans le fichier `~/.netrc` (ligne vide indispensable en fin d'entrée)

```
machine ftp.loria.fr
login anonymous
passwd toto@unice.fr
macdef init
cd /pub/rep/truc
binary
prompt
get fichier1
cd /pub/rep2/bidule
mget fichiert*
quit
```

3. Lancer

```
at -f ~/reptransftp/ftp.at 04:00 January 29
```

4.8.3 Comparaison de rcp et de ftp

D'une manière générale ftp est plus souple que rcp, surtout si la copie s'effectue entr deux utilisateurs différents. ftp permet de travailler entre deux machines qui n'ont pas même système d'exploitation. Même si l'on travaille sur deux systèmes Unix compatible rcp oblige à avoir un fichier "`.rhosts`" qui doit contenir les lignes nécessaires à l'acc distant.

Cependant, au contraire de rcp, la plupart des versions actuelles de ftp ne permettent pas de copier une arborescence entière en désignant le répertoire racine. Cet inconvénien n'existe plus sur les dernières versions de ftp. De plus, rcp permet de se passer d'entrer u mot de passe, ce qui peut être utile si on veut lancer une copie dans un shellscript, sa intervention de l'utilisateur.

4.8.4 Traitement des fichiers récupérés

Les fichiers copiés sont souvent compactés avec la commande *compress*. On reconna les fichiers compactés à leur extension "`.Z`". On peut les décompacter avec la comman *uncompress* :

```
uncompress fichier.Z
```

La commande *zcat* décompacte un fichier compacté par compress et envoie le résulta sur la sortie standard, ce qui permet de le visualiser directement ou de traiter le résulta avec une autre commande placée après un pipe. On peut par exemple décompacter u fichier et le "déplier" par *tar* en une seule commande :

```
zcat fichier.tar.Z | tar xf -
```

L'extension ".gz" indique un fichier compacté avec la commande *gzip*. On peut les décompacter avec l'option "`-d`" de gzip :

```
gzip -d fichier.gz
```

L'option "`-c`" ou la commande *zcat* permettent de déccompacter un fichier sur la sortie standard pour l'afficher ou l'envoyer vers un pipe (pour l'imprimer par exemple).

Les fichiers qui ont une extension ".tar" sont à traiter avec la commande tar. Un fichier ".tar" peut correspondre à une arborescence complète "repliée" en un seul fichier. Avant de le "déplier" il est prudent de lister son contenu avec la commande

```
tar tvf fichier.tar
```

ne serait-ce que pour savoir si les noms sous lesquels les fichiers ont été enregistrés (et sous lesquels ils seront recopiés sur la machine locale) sont "absolus" (débutent par un "/") ou relatifs. On évite ainsi le risque d'écraser des fichiers.

On tape ensuite la commande :

```
tar xvpf fichier.tar
```

pour déplier le fichier ".tar" (l'option "`p`" évite les problèmes liés au "umask").

L'étape suivante est de lire les fichiers "`README`" pour savoir comment mettre en marche (ou compiler) le logiciel copié. La plupart du temps il suffit (!) de personnaliser quelques fichiers et de lancer une ou plusieurs commandes *make*.

On peut aussi récupérer des fichiers archivés sous forme de shellscripts appelés "archives shell" (ou fichiers "*shar*"). Chaque fichier archive contient plusieurs fichiers sources. Le mode d'emploi pour extraire ces fichiers sources est souvent écrit en commentaires au début du fichier archive. Il suffit de lancer un shell avec ces fichiers en entrée ; par exemple, si on travaille avec le shell *ksh* (de toute façon, la première ligne des fichiers archives shell est souvent "`#! /bin/sh`" et le shell qui sera utilisé sera donc *sh*) :

```
ksh fichier-archive.shar
```

On crée ainsi un ensemble de fichiers sources que l'on doit ensuite traiter, comme pour les fichiers récupérés d'un *tar*.

4.8.5 Autres moyens de copier des fichiers distants

Si on travaille sous X Window, on peut utiliser les commandes *xftp* ou *mftp* (voir cours X Window) qui offrent les fonctions de *ftp* avec une interface plus conviviale.

emacs (voir cours Unix) permet de copier des fichiers distants dans le mode "dired" (emacs utilise en fait ftp).

Tous les logiciels de recherche des informations dans Internet (étudiés en 5.3) offrent la possibilité de transférer les informations trouvées (ces outils utilisent en fait ftp).

On peut aussi transmettre des fichiers par la commande *mail*. C'est la seule solution si une des machines est sur un réseau qui n'est que partiellement connecté au réseau Internet et qui ne peut échanger que du courrier éléectronique avec les machnines d'Internet, ou si on ne peut disposer au moment de la copie des mots de passe des deux comptes qui veulent

s'échanger des fichiers (pour autoriser les copies).

Exemple 4.10

```
mail toto@egrcvx.bitnet < fichier
```

Pour transmettre un fichier exécutable binaire, il faut le modifier au préalable avec l commande *uuencode* qui envoie le fichier codé sur la sortie standard :

```
uuencode fichierlocal nomdestination | mail toto@egrcvx.bitnet
```

pour le transmettre "en 7 bits" dans le réseau. Le destinataire doit sauvegarder le messag reçu (par exemple avec la commande "s" si on lite le message avec la commande *mail* Le fichier sauvegardé (par exemple sous le nom "message"est ensuite repassé en "en 8 bits par la commande *uudecode* ; *uudecode* enlève automatiquement l'en-tête du mail et crée u fichier du nom donné par la commande *uuencode* (nomdestination pour l'exemple donné)

```
uudecode message
```

Sur Macintosh, le programme *Binhex* a les mêmes fonctionnalités que *uuencode* et *u decode*.

La commande *kermit* permet aussi la copie de fichiers distants. Elle est surtout utilise dans les connexions par modem (voir 7.3.3).

Voici un tableau des extensions les plus fréquemment rencontrées et des types de fichie associés :

.tar	traité par tar
.Z	compacté par compress
.gz	compacté par gzip
.uu	traité par uuencode
.shar	shellscript archive
.zip	traité par pkzip (PC)
.hqx	traité par binhex 4.0 (Macintosh)
.bin	traité par binhex 5.0 (Macintosh)
.sea	fichier auto-extractible en cliquant dessus (Macintosh)
.sit	compacté par Stuffit (Macintosh)
.cpt	compacté par Compact Pro (Macintosh)

4.9 Informations sur le réseau

4.9.1 Utilisateurs du réseau (finger, rwho)

`finger [@machine]`

affiche les informations de la commande *who* et des informations informelles sur les utili sateurs (leur nom et prénoms, leur téléphone,...) actuellement connectés sur une *machi* (la machine locale par défaut).

`finger utilisateur[@machine]`

affiche les informations sur un *utilisateur* même s'il n'est pas connecté. finger donne plu d'informations sur les utilisateurs que lorsque l'on n'indique pas d'utilisateurs en particulie De plus les contenus des fichiers `.plan` et `.project` des répertoires HOME des utilisateu

sont affichés. Ces deux fichiers contiennent en général des informations sur le lieu de travail et les projets en cours de l'utilisateur.

`rwho [-h] [utilisateurs]...`

affiche les noms, terminaux d'accès, heure de login des *utilisateurs* connectés au réseau local de la machine où est lancé la commande. Les utilisateurs sont triés par nom ou par machine si l'option -h est donnée.

Elle n'est pas disponible sur le réseau de l'université car le daemon *rwhod* n'y est pas lancé (il occasionne trop de charge sur le réseau).

4.9.2 Machines du réseau (ping, ruptime)

`ping [-s] machineD`

indique si machineD est bien connectée au réseau. ping est le plus souvent situé sous `/etc` ou sous `/usr/etc` et il faut donner son chemin absolu s'il n'est pas indiqué dans la variable `PATH`.

Option

`-s` donne en plus des indications sur le temps de connexion à la machine distante. Ce peut être utile pour voir s'il n'y a pas de problème sur le réseau ou pour repérer les serveurs les plus proches si on veut copier des logiciels distants.

`ruptime`

affiche la charge des machines du réseau local de la machine où est lancé la commande. Sur le réseau de l'Université de Nice, elle ne donne aucune indication car elle obtient ses informations du daemon rwhod.

La commande *uptime* affiche la charge de la machine locale (sur laquelle a été lancée la commande). On peut l'utiliser avec rsh (voir 4.7) pour avoir la charge d'une machine distante sur laquelle on a un compte :

```
$ rsh machineD uptime
```

4.10 Chemin suivi par les messages traceroute)

La commande *traceroute* cherche à découvrir le chemin suivi par les messages envoyés à une machine. Elle peut permettre de découvrir les raisons d'un mauvais fonctionnement dans un réseau.

Elle est réservée à l'administrateur du système.

Chapitre 5

Services pour l'utilisateur final

Au contraire des services décrits au chapitre 4, qui sont encore des commandes d même niveau que les commandes de base d'Unix (copie de fichiers, lancement de processus les services décrits dans ce chapitre offrent des services complets à l'utilisateur final no informaticien : échange de messages avec d'autres utilisateurs, recherche d'informations.

Ce chapitre donne moins de détails sur les commandes que le chapitre 4. Ces con mandes sont assez simples à utiliser (surtout dans l'environnement X Window). Le lecte pourra trouver des informations complémentaires dans les pages du manuel de référenc (commande *man*) ou dans le livre de Ed Kroll cité dans l'introduction.

Remarque 5.1

Toutes les adresses Internet citées dans ce chapitre sont données sous toute réserv car elle sont susceptibles d'être modifiées à tout moment.

5.1 Courrier électronique

mail est la commande standard pour envoyer et recevoir du courrier.

Le service du courrier électronique utilise le protocole SMTP. Ce protocole utilise serveur de nom, gère la bonne arrivée du courrier.

Outre la commande *mail*, on peut envoyer et recevoir du courrier électronique en ut lisant d'autres logiciels : *emacs*, *mh* (*xmh* ou *exmh* sous X Window) et *netscape* sont l plus utilisés. Il faut savoir que ces systèmes sont plus ou moins incompatibles entre eu en particulier dans leur manière de conserver les messages déjà lus. Il vaut mieux faire u choix et s'y tenir.

emacs offre un mode "mail" qui facilite la saisie des messages et offre quelques fon tionnalités de plus que mail.

Le système "MH" (*Rand Message Handling*) offre de nombreuses facilités pour la gestio des mails. Au contraire des commandes *mail* et *emacs* pour lesquelles la gestion du courri est traité par une seule commande, le système MH est formé de nombreuses command qui permettent d'écrire des messages, d'en lire, d'en ranger dans des fichiers, etc. Sous

Window, la commande *xmh* offre toutes les fonctionnalités du système MH en une seule commande et avec une interface graphique plus conviviale.

Le navigateur Web de Netscape offre une gestion simple, conviviale et puissante du courrier électronique. Il permet de gérer facilement un carnet d'adresse et des dossiers de messages. Les relations avec le Web sont facilitées. La dernière version (*Communicator*) offre même la possibilité de filtrer automatiquement les messages à l'arrivée pour les répartir dans des dossiers ou sous-dossiers.

5.2 Conversation sur le réseau

5.2.1 Conversation à deux : talk

`talk` *utilisateur* `[`*terminal*`]`

affiche, sur le terminal où est entré l'*utilisateur*, le message "Message from..." suivi de la commande à taper pour répondre. Si l'*utilisateur* est entré sur plusieurs terminaux, on peut spécifier le *terminal* où s'affichera le message.

Lorsque l'*utilisateur* répond, les deux interlocuteurs ont chacun deux fenêtres séparées sur lesquelles ils peuvent envoyer et recevoir du texte. Pour arrêter il leur suffit de taper le caractère d'annulation de commande (Ctrl C le plus souvent).

talk est construit au-dessus du protocole UDP.

Il n'est pas toujours agréable d'être interrompu dans son travail par un talk. On peut se protéger en tapant "`mesg no`" ; on peut autorisé à nouveau les talks en tapant "`mesg yes`".

Remarque 5.2

Cette commande perturbe le travail des utilisateurs et il est préférable d'utiliser le courrier électronique si le message n'est pas urgent.

5.2.2 Conversation à plusieurs : IRC

IRC (*Internet Relay Chat*) est un réseaux de serveurs qui permettent aux utilisateurs des programmes clients IRC de se joindre à des groupes de discussions (*channel* dans le jargon IRC).

On commence par se connecter à un serveur. Tous les serveurs sont connectés entre eux ; il faut donc choisir le plus proche. On peut attendre parfois longtemps ou mettre être rejeté si le serveur est surchargé. Pour se connecter au serveur irc.Eurecom.fr :

```
irc server irc.Eurecom.fr
```

On tape ensuite la commande "/help" pour en savoir plus. Si on tape "intro" quand on est dans "help" on obtient des informations pour les débutants.

Toutes les commandes commencent par "/" ; tout ce que vous tapez et qui ne commence pas par "/" est considéré comme du texte que vous envoyez dans le groupe que vous avez choisi. Les noms de groupes commencent par le caractère "`#`".

Voici quelques commandes essentielles :

- `/list` affiche la liste les groupes de conversation (`/list -wide -min 5` pour affiche sous un format compact tous les groupes qui ont au moins 5 utilisateurs),
- `/join #nom-groupe` permet de se joindre à un de ces groupes (on peut participer plusieurs groupes en même temps),
- `/who #nom-groupe` affiche les surnoms des participants d'un groupe,
- `/nick surnom` permet de se choisir un surnom,
- `/leave` quitte le groupe,
- `/quit` quitte IRC.

5.3 Recherche d'informations sur le réseau Internet

Les réseaux informatiques offrent une quantité immense d'informations qui concerne tous les domaines. Chaque jour des nouveaux serveurs sont ajoutés ou enrichis. Pour l'ut lisateur, le plus difficile est souvent de savoir où il peut trouver des informations qui pou raient l'intéresser.

L'utilisateur dispose d'outils pour rechercher les informations. Nous allons étudier l outils les plus utilisés. Ils fonctionnent tous en mode "client-serveur" (voir 4.2). Nous n donnerons pas les détails d'utilisation mais plutôt les fonctionnalités offertes. Le lecte intéressé par une fonctionnalité recherchera dans le manuel en ligne comment opérer.

5.3.1 Archie

Le système "Archie" a pour but de répertorier tous les serveurs qui offrent un "ft anonymous" dans le monde (voir 4.8.2). On peut ainsi lister tous les ordinateurs qui offre *emacs* et récupérer le code de *emacs* depuis une de ces machines.

Les serveurs "Archie" sont des ordinateurs dans lesquels on a indexé les logiciels fichiers disponibles sur les serveurs "ftp anonymous". Cette information est enregistr dans une base de données remise à jour régulièrement (environ chaque mois).

On peut accéder à ces serveurs par un client archie. On peut aussi se connecter direct ment aux serveurs par telnet avec "archie" comme nom de login. Voici une liste de serveu Archie avec leur localisation :

archie.univ-rennes1.fr	France
archie.sura.net	USA
archie.unipi.it	Italie
archie.th-darmstadt.de	Allemagne
archie.doc.ic.ac.uk	Angleterre

La commande *xarchie* offre de plus les fonctionnalités de *ftp* pour récupérer les fichier

Malheureusement, le système "Archie" comporte des lacunes. Les serveurs Archie ne sont pas très nombreux et les temps d'attente peuvent être longs pendant les consultations. On peut même être rejeté s'il y a déjà trop d'utilisateurs connectés. Les informations contenues dans leurs bases peuvent être provisoirement erronées entre les différentes mises à jour. Ce qui est plus gênant, de nombreux serveurs ftp ne sont pas répertoriés par les serveurs Archie.

Les utilisateurs qui récupèrent souvent des fichiers sur le réseau ont donc intérêt à se constituer leur propre liste de serveurs "ftp anonymous" et de n'utiliser Archie (ou les recherches sur le Web étudiées ci-après) que s'ils n'ont pas trouvé ce qu'ils cherchent dans leurs serveurs habituels.

Il existe des clients Archie pour de nombreux types de machines et en particulier pour Unix, X Window, PC et Macintosh. Le nom du programme est *archie* sous Unix en mode "texte" et *xarchie* sous X Window.

On peut aussi utiliser le système Archie avec le courrier électronique. On envoie un message contenant une commande "archie" (voir "`man archie`") pour demander la recherche de fichiers et on reçoit en retour un message donnant les endroits où se situent les fichiers.

5.3.2 Gopher

Le système "Gopher" permet de rechercher des informations sur le réseau en étant piloté par des menus.

Ces menus sont fournis par des serveurs "Gopher" auxquels on se connecte grâce à des clients ou en se connectant directement avec telnet. Il existe des serveurs "publics" auxquels tous les utilisateurs peuvent se connecter par une commande *telnet*. Ces serveurs publics ne permettent pas de transférer par ftp les informations trouvées (on peut seulement parcourir leurs menus pour consulter des informations et les transférer par la commande mail). Si l'on n'a pas un compte sur une machine sur laquelle tourne un serveur Gopher, il est préférable de travailler par l'intermédiaire d'un client Gopher afin de transférer directement les informations trouvées par ftp.

Un serveur public est le serveur "gopher.ebone.net" auquel on peut se connecter avec le nom de login "gopher". "`gopher.unice.fr`" est le serveur de l'université de Nice Sophia-Antipolis. On peut y accéder en lançant le programme *xgopher*.

Chaque serveur Gopher possède ses propres menus de recherche. Si l'on veut les noms des serveurs Gopher qui offrent les informations que l'on recherche, on peut utiliser le service "Veronica" (*Very Easy Rodent-Oriented Net-Wide Index to Computerized Archives*). Ce service est le plus souvent accessible par un des menus des serveurs Gopher. Il permet la recherche par mots-clés. Veronica indexe périodiquement les intitulés des menus des serveurs Gopher (sont ainsi indexé un nom de fichier ou les mots inclus dans un des choix des menus). Une recherche Veronica peut comporter des opérateurs booléens et une option "-t" peut limiter la recherche à un type de service. Comme avec Archie, tous les serveurs ne sont pas indexés par Veronica.

Un système de "signets" permet à l'utilisateur de se constituer son propre menu avec les serveurs qu'il consulte le plus souvent.

Les menus de Gopher peuvent offrir l'accès au système "Wais", avec certaines limitation en particulier sur les types de documents visualisables (le système Wais est étudié en 5.3.3 Des choix de menus offrent aussi la possibilité de se connecter à une machine par telnet.

Chaque choix de menu a un type affiché sur la même ligne que l'intitulé du choix. Voi les principaux types :

.	(un point ou rien du tout) fichier (que l'on peut transférer ou con
/	répertoire (ou sous-menu)
<?> (ou <idx>)	recherche par index
<CSO>	recherche d'information sur une personne ("pages blanches")
<TEL>	telnet sur une machine

Le problème avec Gopher est que chaque serveur a ses propres menus et qu'aucun norme n'est universellement admise pour les découpages en menus et les intitulés des choi des menus. Les mêmes informations sont rangées dans des menus différents sur les dive serveurs.

Gopher n'est plus vraiment utilisé depuis l'arrivée du Web (voir 5.3.4).

Il existe des clients Gopher pour de nombreux types de machines et en particulier pou Unix, X Window, PC et Macintosh. Le nom du programme est *gopher* sous Unix en mod texte, *xgopher* sous X Window.

5.3.3 Wais

Wais est l'acronyme de *Wide Area Information Service*. Wais est particulièreme adapté à la recherche documentaire. Le système Wais permet la recherche d'informatio d'après le *contenu* et le descriptif des fichiers enregistrés. Les contenus de ces fichiers doive être préalablement indexés. Le programme qui effectue l'indexage peut traiter des fichie de nombreux formats : texte simple en ASCII, LaTeX, dvi, Postscript, etc.

La recherche des documents s'effectue en deux étapes. L'utilisateur choisit tout d'abo les serveurs ("source" dans le vocabulaire Wais) dans lesquels il recherchera des document Pour cela, on utilise un serveur nommé `directory-of-servers.src` qui contient la lis des serveurs disponibles. Quand on a choisit une ou plusieurs sources, on fait ensuit une recherche par mots-clés pour sélectionner les fichiers que l'on souhaite consulter o transférer. Une source Wais correspond à une bibliothèque qui contient des informatio indexées et enregistrées localement. On remarquera que ce concept est différent des concep de serveurs des autres systèmes de recherche pour lesquels un serveur correspond à u machine qui contient des informations locales mais aussi des accès à des informatio enregistrées sur d'autres machines.

Les possibilités offertes à l'utilisateur pour exprimer sa recherche étaient très restreint dans les premières versions des programmes utilisant Wais. La situation s'améliore avec l nouvelles versions. Wais utilise des normes de recherche dans les bases de données (norm Z39.50 par exemple).

Les fichiers trouvés sont classés par ordre décroissant des nombres d'occurrences d mots-clés trouvés dans les fichiers. Ils reçoivent une "note" qui est supposée indiquer l'ad quation du fichier au critère de recherche. Le premier fichier reçoit la note 1000. Si o

indique plusieurs mots-clés (séparés par un espace) pour la recherche, la recherche se fera sur chacun des mots-clés pris individuellement. Dans les versions encore les plus courantes, la recherche ne permet pas d'indiquer que deux mots-clés doivent se suivre ou être proches ni d'indiquer qu'ils doivent appartenir à une certaine partie du texte (par exemple les sections réservées au résumé ou aux auteurs). La recherche fournit ainsi quelquefois des fichiers qui ne correspondent pas à ce que recherchait l'utilisateur. Par exemple, les deux mots-clés "calcul formel" vont sélectionner des fichiers qui contiennent souvent le mot "calcul" seul. Wais demande une bonne expérience dans la façon de donner les mots-clés.

L'utilisateur peut afficher les documents trouvés si la machine dispose des programmes de visualisation correspondant au type des fichiers trouvés (par exemple *xdvi* pour les fichiers de type "dvi"). Il peut aussi les sauvegarder sur sa machine.

On peut accéder à un serveur Wais directement par telnet ou par l'intermédiaire d'un client Wais. "info.funet.fi" est un serveur public auquel on peut se connecter par telnet avec le login "wais".

Il existe des clients pour toutes les machines communément utilisées et en particulier pour Unix, X Window, PC et Macintosh. Le nom du programme est *swais* en mode texte sous Unix et *xwais* sous X Window.

5.3.4 WWW (ou W3)

WWW (*World Wide Web*; on dit aussi W3 ou "le Web") est l'outil le plus récent pour rechercher des informations sur les réseaux informatiques. Le mot anglais "*web*" signifie "toile d'araignée"; il fait référence au réseau mondial formé par les ordinateurs reliés par Internet.

Les clients Web (navigateurs, ou *browsers* ou butineurs) permettent de se connecter à des serveurs et de passer très facilement d'un serveur à l'autre pour rechercher une information.

La recherche se fait sur le principe de l'*hypertexte*: le texte affiché par un serveur WWW contient des mots (ou portions d'écran) spéciaux, les "liens" (*link* en anglais). Si l'utilisateur clique avec la souris sur ces liens (ou tape le numéro du lien s'il travaille avec une interface en mode texte), il lance une exécution associée à ces liens. WWW est multi-média; un lien peut correspondre à l'affichage d'un texte ou d'une image, à l'audition d'un document sonore ou même au déroulement d'une séquence animée, et tout ceci à l'intérieur d'un seul document composite. Comme pour le système Wais, la machine locale doit disposer des programmes de visualisation correspondant au type des fichiers trouvés.

Les liens d'hypertexte peuvent faire référence à des objets locaux au serveur ou situés sur un autre serveur WWW du réseau. On est dans ce dernier cas connecté automatiquement au serveur Web distant.

Les textes écrits pour être consultés avec un client W3 sont écrits dans le langage html. Ce langage est simple. Il est simple d'ajouter des documents à un serveur Web et pas trop difficile de créer son propre serveur Web (il existe des versions gratuites des logiciels nécessaires au montage d'un serveur Web). De plus les clients W3 sont très facile à utiliser.

Pour ces raisons, W3 a connu un énorme succès et a contribué à la croissance exponentiel d'Internet ces dernières années.

WWW offre de plus l'accès à la plupart des autres systèmes: Wais, Gopher, Veronic ftp, telnet, News, etc. Il a l'avantage de présenter une interface uniformisée pour tous l systèmes. Cependant, si ces systèmes sont parfois plus agréables à utiliser avec WW pour les fonctionnalités simples (par exemple, les News sont prétraitées pour WWW l'on dispose de liens d'hypertexte pour passer d'un article à l'autre), WWW n'offre génér lement pas toujours toutes les fonctionnalités de ces systèmes. Il reste avant tout un ou de recherche. Par exemple, si l'on sait exactement où trouver des fichiers, il est meille d'utiliser ftp plutôt que WWW pour les transférer car, en particulier, WWW ne perm de transférer qu'un seul fichier à la fois et il n'affiche pas la taille des fichiers. Le systè WWW bénéficie de nombreux développement; on peut donc espérer que ces lacunes sero comblées dans les futures versions.

Les échanges entre un client et un serveur Web utilisent le protocole http. WWW utili le concept de URL (*Uniform Resource Locator*) pour repérer les différentes ressources qu utilise. Ces ressources peuvent être de plusieurs types qui correspondent à divers protocole Voici le format des URL pour les différents types:

Fichier local:	file:*chemin-fichier*
Ftp:	ftp://*machine*/*chemin-fichier*
Telnet:	telnet://*machine*
Gopher:	gopher://*machine*
Wais:	wais://machine/serveur
Hypertexte:	http://*machine*/*chemin-fichier*
News:	news:*groupe* (pas de "//"!)

On peut accéder à un serveur Web directement par telnet ou par l'intermédiaire d'u client Web. Un serveur public est "telnet.w3.org" auquel on peut se connecter par telnet. I serveur Web de l'université de Nice Sophia-Antipolis a pour URL "`http://www.unice.fr`" Pour y accéder directement, on peut lancer un client W3 avec cet URL comme paramètr

Il existe des clients pour toutes les machines communément utilisées et en particuli pour Unix, X Window, PC et Macintosh. Le navigateur "*netscape*" de Netscape est le pl utilisé. Il existe en versions Unix, Windows et Macintosh. Sur PC il est concurrencé par navigateur "*Internet Explorer*" de Microsoft (qui n'existe que sous Windows).

Pour se faciliter l'accès à ses adresses préférées, l'utilisateur peut se constituer s propres menus avec les adresses qu'il a l'habitude de consulter (signets ou "*bookmarks*").

5.4 News

Le système USENET (*User's Network*) permet aux utilisateurs de partager des info mations sur le réseau, de poser des questions et de recevoir des réponses.

Ce système est mondial et il offre une source énorme d'informations de tous genre scientifiques, littéraires, artistiques, etc.

Le réseau USENET est composé de centres qui, en échange d'un abonnement annue

peuvent récupérer les informations du réseau et les mettre à la disposition des utilisateurs de leurs réseaux. Les transferts utilisent le système uucp (*Unix to Unix copy*) ou passent par TCP/IP.

Les informations sont classées en groupes, les *newsgroups*, qui sont eux-mêmes ordonnés hiérarchiquement. Les noms complets des groupes sont constitués des noms des groupes auxquels appartiennent les groupes ordonnés suivant les niveaux (les noms des groupes sont séparés par des points). Par exemple, "comp.unix.questions".

Les principaux groupes de premier niveau sont :

comp	informatique (computing)
news	le réseau des news lui-même
rec	récréations
sci	sciences
soc	social
talk	discussions
alt	"alternatif"
fr	pour des groupes français

Le groupe "unice" a été créé localement pour les sujets traitant de l'université de Nice.

Quelques groupes terminaux :

comp.unix.questions
comp.lang.c
comp.emacs
sci.math
alt.boomerang (!)
unice.unix
essi.ragots...

Les informations sont données sous forme d'articles dans chaque groupe.

La plupart de ces groupes sont d'accès libres : chacun peut y poser ses propres questions et répondre aux questions déjà posées. Les News sont ainsi encombrées de beaucoup d'articles sans intérêt ou même faux. Il n'est pas toujours facile d'y trouver ce que l'on cherche. Il est recommandé de consulter tout d'abord les "FAQ" (*Frequently Asked Questions*) lorsque l'on aborde un nouveau groupe. Ce sont les questions les plus fréquemment posées sur les News avec les bonnes réponses. Ces FAQ sont régulièrement postées dans les groupes qu'elles concernent. On peut aussi les trouver aux adresses
`http://www.lib.ox.ac.uk/internet/news/`
ou
`http://www.cis.ohio-state.edu/hypertext/faq/usenet/FAQ-List.html`.
Le moteur de recherche *Infoseek* (`http://guide.infoseek.com/`) permet aussi de rechercher des mots-clés dans les FAQ. Les FAQ se trouvent aussi sur le serveur FTP `rtfm.mit.edu` sous le répertoire `pub/usenet/news.answers` (taper `get index` pour avoir la liste des FAQ disponibles).

Il existe de nombreux clients pour consulter les news, sur PC, Macintosh ou Unix. Si on dispose de l'interface X Window, la commande *xrn* est simple à utiliser. Le logiciel "Web" *netscape* (voir 5.3.4) offre de nombreuses facilités comme l'affichage automatique

des images ou le classement des messages suivant les "*threads*" (les réponses sont classé
avec les questions).

Sous Unix, il peut être nécessaire d'affecter à la variable d'environnement `NNTPSERVE`
le nom de la machine serveur de news que l'on utilise. Par exemple,

```
NNTPSERVER=news.unice.fr
export NNTPSERVER
```

Tous ces programmes utilisent le fichier `.newsrc` du répertoire HOME de l'utilisateu
Ce fichier contient les noms des groupes auxquels l'utilisateur s'est abonné (un "!" suit
nom des autres groupes) et les articles qui ont déjà été lus dans chaque groupe.

Le flux d'information est trop important pour que les articles soient conservés indéf
niment sur les serveurs de news. Certaines machines archivent cependant tous les articl
d'un certain groupe des news. Pour connaitre ces machines, l'utilisateur peut, par exempl
rechercher le nom du groupe avec le système *archie* (voir 5.3.1).

5.5 Accès aux bases de données

La connexion aux réseaux offre de nombreuses possibilités de consultations de bases d
données. Certaines sont payantes comme la base de donnée scientifique "PASCAL", ou
base de données "DIALOG" sur les entreprises. On peut s'abonner ou payer pour chaqu
interrogation (au temps, ou au volume des questions ou des réponses).

Les systèmes décrits en 5.3 donnent accès à de nombreuses bases de données gratuite
(en particulier des bases de données bibliographiques).

5.6 Listes de diffusion (ou de messagerie)

L'utilisateur peut s'inscrire à une liste de diffusion de son choix. Par l'intermédiaire d
courrier électronique il participera alors aux débats qui animent ces listes et recevra l
informations qui y sont diffusées.

Le serveur "Web" de l'université donne une liste de nombreuses listes de diffusion ave
leurs thèmes.

Le plus souvent pour avoir des informations sur une liste il suffit d'envoyer un me
sage sans sujet et contenant le seul mot "HELP" à l'adresse "listserv@*machine*", où *m*
chine désigne la machine où est hébergée la liste. Par exemple, si le nom de la liste e
"`www-fr@univ-rennes1.fr`", on enverra ce message à "`listserv@univ-rennes1.fr`".

En retour on reçoit en particulier les thèmes abordés dans la liste et la commande pou
s'abonner à la liste. L'ordre est souvent du type
"`subscribe` *nom-liste vos-coordonnées*"
mais cela dépend de la liste. Pour l'exemple cité ci-dessus, le message pourrait être

```
subscribe www-fr Pierre Dupond Universite de Nice
```

Le site Web `http://www.neosoft.com:80/internet/paml/indexes.html` donne u
liste des listes de diffusion. `http://tile.net/` permet de rechercher des listes ou d

groupes de News suivant un mot-clé. Le site Web `http://www.cru.fr/listes/` donne une liste des listes françaises et facilite l'abonnement et l'accès aux archives de ces listes.

Lorsque l'on s'abonne à une liste il faut s'attendre à recevoir des dizaines de messages par jour. Pour éviter cela on peut se contenter de consulter régulièrement les archives de la liste. Les messages y sont le plus souvent classés par thèmes et un outil de recherche par mots clés y est souvent disponible. On ne s'abonnera à une liste que si on veut participer aux débats. Si on veut poser exceptionnellement une question, il est plus simple de la poser dans les "*news*".

5.7 Stratégie pour une recherche sur le Web

Le Web est le plus utilisé de tous les outils disponibles sur Internet. La somme d'information gigantesque disponible sur le Web déroute souvent le débutant qui ne sait comment retrouver ce qui l'intéresse parmi toutes ces informations. Cette section donne quelques conseils et informations pratiques.

La première chose à faire est de parcourir ses serveurs Web locaux sur lequel des liens ont déjà été établis vers les domaines d'intérêt des utilisateurs.

Durant ses recherches sur le Web on n'hésitera pas à repérer quelques sites par des signets personnels pour les retrouver plus tard plus facilement. Dès que les signets sont trop nombreux, il faut structurer ses signets en dossiers (consulter le mode d'emploi des navigateurs).

Quelques adresses du domaine "unice"(ajouter "`http://`" devant toutes les adresses Web) :

- Université de Nice : `www.unice.fr/`
- Étudiants du MIPS (Deug, licence, maîtrise, IUP Miage) : `www-mips/`,
- Département de mathématiques : `math.unice.fr/`,
- Liste de moteurs de recherche : `nephi.unice.fr/html/Search/`

Si l'utilisateur ne trouve pas ce qu'il veut, il devra s'aider de moteurs de recherche. La technique des moteurs de recherche est de parcourir sans fin le Web par des programmes automatiques et d'indexer tout (ou presque tout) ce qu'ils rencontrent. Certains classent ensuite ces informations par thèmes. Les responsables de serveurs Web peuvent aussi déclarer aux moteurs de recherche les pages qu'ils hébergent sur leur site.

Dans les sections suivantes nous allons étudier les différentes catégories de moteurs de recherche :

- moteurs pour les recherches géographiques,
- moteurs pour les recherches par thèmes,
- moteurs pour les recherches par mots-clés,

Pour chaque catégorie de moteur, l'utilisateur choisira son moteur de recherche préfé et n'ira voir les autres que lorsque la première recherche n'aura rien donné.

On peut aussi rechercher l'adresse électronique d'une personne.

5.7.1 Recherche géographique

On utilise ce type de recherche quand on recherche un site situé dans une région d monde. Par exemple, si on veut choisir un hôtel à Pékin pas trop éloigné d'une certai université, on cherche un serveur à Pékin pour obtenir une carte de Pékin avec les hôtel les universités et, si possible, les stations de métro.

Voici des adresses intéressantes de moteurs de recherche géographique :

- *Virtual Tourist* (pour le monde) : `www.vtourist.com/webmap/`,
- Serveurs européens : `www.tue.nl/europe/`,
- UREC (CNRS) pour les serveurs français : `www.urec.fr/France/`

5.7.2 Recherche par thème

Ce type de recherche est utile quand on est intéressé par tout ou une grande part de ce qui concerne un thème et pas seulement par un point, un article ou un logiciel bie particulier. C'est aussi intéressant lorsque l'on recherche une information sans avoir u idée bien précise sur l'outil ou la théorie qui conviendra et que l'on connait pas les mots-cl assez précis qui permettraient une recherche par mot-clé efficace.

On recherche par exemple des informations sur la façon d'accéder à des bases de donné avec le langage Java, ou tout sur son chanteur préféré.

Les moteurs thématiques classent les informations du Web suivant une arborescence thèmes. On arrive ainsi à des thèmes assez pointus en partant de thèmes généraux. Le pl souvent on peut aussi effectuer des recherches par mots-clés comme il est décrit ci-dessou en ayant la possibilité de restreindre la recherche à un thème précis.

Quelques adresses :

- *Yahoo!* est le plus connu et sans doute l'un des meilleurs : `www.yahoo.com/` ; versio française en `www.yahoo.fr/`
- Nomade (recherche sur sites francophones) : `www.nomade.fr/`,
- *Infoseek* : `guide.infoseek.com/`,
- *Webcrawler* : `webcrawler.com/`

5.7.3 Recherche par mot-clé

De nombreux moteurs de recherche offrent la possibilité d'effectuer une recherche rapide par mot-clé.

Par exemple on peut chercher le site de l'association GUTenberg ou des articles qui traitent du café en Côte d'Ivoire. Pour cela on donnera au moteur de recherche des mots-clés, éventuellement liés par des opérateurs logiques comme "ou", "et", "sauf" (la plupart du temps en anglais !) et le moteur de recherche fournit des références sur ces mots-clés, classées par ordre de meilleure correspondance.

Ces recherches sont souvent difficiles car il faut bien choisir ses mots-clés sous peine d'être submergé par des centaines de mauvaises références.

Malgré tout, c'est la meilleure solution si l'on veut un éventail de tout ce qui est sur le Web sur un point précis.

Quelques adresses :

- *Alta Vista* est incontournable. Il possède des millions de références : `altavista.digital.com/` . La version française (on choisit sa langue et son pays et on met un signet sur le site obtenu) se trouve en `www.altavista.telia.com/`
- *Lycos* français : `www-fr.lycos.com/`

5.7.4 Recherche d'une adresse électronique

Il n'existe pas d'annuaires officiels pour les adresses Internet comme pour le téléphone. C'est impossible étant donné la croissance d'Internet et la mobilité des utilisateurs.

On peut cependant essayer de rechercher l'adresse "e-mail" d'un utilisateur d'Internet grâce à des moteurs de recherche, et on obtient souvent satisfaction. La recherche s'effectue par le nom, l'adresse, le numéro de téléphone, le domaine Internet, etc. de l'utilisateur.

- *Whowhere* français : `www.french.whowhere.com/`,
- *Four11* : `www.four11.com/`,
- *Internet adress Finder* : `www.iaf.net/`
- Annuaire Renater (réseau des universités françaises) : `www.univ-valenciennes.fr/CRU-annuaire.html`

Pour les adresses du personnel de l'université de Nice, il est plus simple de taper "`ph` *nom-recherché*" sur la machine samoa.

5.7.5 Serveurs spécialisés dans la distribution de logiciels

Nous donnerons ici quelques serveurs sur lesquels on peut récupérer des logiciels en *freeware* (entièrement gratuits) ou en *shareware* (que l'on peut essayer avant d'envoyer une certaine somme, souvent modique, à celui qui a écrit le programme) :

- *Jumbo!* : `jumbo.com/`,

- Partagiciel (logiciels français) : `www.partagiciel.com/`,
- Pour Windows 95 : `www.windows95.com/`,
- *Shareware Junkies* : `www.sharewarejunkies.com/`,
- *Shareware!* : `www.shareware.com/`.

5.8 Récupérer des sites Web

Quelques logiciels gratuits permettent d'automatiser la récupération sur sa propre ma chine des documents du Web écrits en plusieurs pages et contenant des images. Voici deu noms de logiciels que vous pourrez récupérer sur Internet : *webcopy* et *wget*.

Ces logiciels ont de nombreuses options qui permettent d'indiquer ce que l'on veu récupérer. Il faut faire très attention de ne pas récupérer trop de pages. Si on chois certaines options, ces logiciels suivent en effet les liens hypertextes des pages de départ ils peuvent ainsi récupérer très (trop) facilement des centaines de méga-octets.

Chapitre 6

Extensions du système qui utilisent le réseau

6.1 Systèmes de fichiers distribués : NFS, RFS, NIS

Deux extensions du système Unix permettent de rendre transparent à l'utilisateur le travail avec des fichiers situés sur une autre machine du réseau. Ces fichiers semblent appartenir à un système de fichiers implanté sur un disque de sa machine locale et rien ne les différencie des autres fichiers dans la syntaxe des commandes : l'utilisateur n'a pas à préciser sur quelle machine ils sont enregistrés, alors que ce n'est pas le cas lorsque, par exemple, il utilise la commande rcp (voir 4.8.1).

Ceci permet en particulier d'offrir aux utilisateurs le même répertoire HOME sur toutes les machines d'un réseau. Cette possibilité est importante au moment où les réseaux de stations de travail se développent : un utilisateur peut se connecter sur n'importe quelle station et retrouver les mêmes fichiers et le même environnement.

Ces extensions sont fondées sur le modèle "client-serveur" (voir 4.2). Une machine serveur accepte de partager ses fichiers avec d'autres machines.

Les deux extensions sont "*Network File System*" (NFS) et "*Remote File System*" (RFS).

NFS peut fonctionner sur un réseau hétérogène dont les machines n'ont pas nécessairement Unix comme système d'exploitation.

RFS ne peut fonctionner qu'avec des machines dont le système d'exploitation est Unix Système V mais il offre quelques fonctionnalités de plus que NFS.

NFS est plus utilisé que RFS. Il est en particulier utilisé à l'université de Nice.

Nous n'entrerons pas dans les détails de mise en œuvre de NFS et de RFS car seuls les administrateurs du système ont des commandes particulières liées à ces systèmes. Les autres utilisateurs n'ont même pas à savoir qu'une partie des fichiers qu'ils utilisent sont "montés" via NFS ou RFS sur leur système de fichiers. Les utilisateurs curieux pourront lancer la commande *df* d'Unix pour savoir quels sont les systèmes montés par NFS sur la machine qu'ils utilisent.

Sachez seulement que si vous recevez un message vous disant que votre répertoire HOME

n'existe plus, c'est certainement qu'il y a eu un problème lors du montage NFS du disqu
qui contenait ce répertoire et que vos données ne sont pas vraiment perdues. NFS utili
en effet le protocole UDP qui n'a pas la fiabilité de TCP et l'administrateur du systèm
peut ne pas s'apercevoir tout de suite d'un problème de montage par NFS d'un disque.

Le système des "*Yellow Pages*", appelé maintenant (pour des raisons de copyrigh
"*Network Information Service*" (NIS) est souvent associé à NFS. Il facilite grandement
gestion des réseaux en centralisant la maintenance des fichiers système tels que les fichie
des mots de passe.

6.2 Interface graphique : X Window

6.2.1 Présentation de X Window

X Window est un système qui permet d'utiliser le multi-fenêtrage dans l'interface entr
la machine et l'utilisateur. Il tire pleinement parti des possibilités offertes par les réseau
informatiques. Il est fondé sur une architecture client-serveur (voir 4.2).

Le serveur X est un programme qui tourne sur la machine locale (station de trava
ou terminal X). Il envoie aux clients X qui utilisent ses services des "événements" q
indiquent les touches du clavier et les boutons de la souris sur lesquels l'utilisateur
appuyé. En retour, il affiche sur l'écran local ce que les clients lui demandent d'afficher pa
des "requêtes". La connexion entre un client X et un serveur se fait au démarrage du clie
X. Si l'environnement est correct, le client X va se connecter avec le serveur X qui tourn
sur la machine locale de l'utilisateur qui a lancé le client X.

Un client X est un programme qui s'exécute sur une des machines du réseau et q
utilise les services d'un serveur X pour effectuer ses entrées-sorties avec l'utilisateur. C
peut être un traitement de texte, un logiciel de comptabilité, un tableur, etc.

X Window est construit autour d'un protocole qui est indépendant du matériel et d
système d'exploitation.

X Window est hautement paramétrable. Il n'impose pas de "look and feel" et de non
breux produits commerciaux ont été construits au dessus de lui : Motif et OpenWindow
sont les plus connus.

X Window est fourni gratuitement à qui le veut (on peut le charger par ftp par exemple
Il est actuellement le standard de fait pour l'utilisation des écrans graphiques. So
importance a crû avec le développement rapide de l'équipement des centres universitair
et d'ingéniérie en stations de travail graphiques et terminaux X.

6.2.2 Terminaux X - stations de travail

Un certain nombre de postes de travail connectés au réseau de l'université de Nice so
des postes serveurs de X Window sans disque appelés "terminaux X". Leurs fonctionnalit
rappellent en effet celles des simples terminaux alphanumériques qui sont connectés à u

ordinateur central et ne font que saisir les données tapées au clavier par l'utilisateur, et afficher les informations envoyées par l'ordinateur central.

Le seul processus qui tourne sur un terminal X est un serveur X[1]. Tous les programmes clients lancés par l'utilisateur sont exécutés sur un ordinateur distant. Le serveur X est le plus souvent chargé au démarrage du terminal X depuis un ordinateur distant en utilisant le réseau. Il est parfois enregistré dans la mémoire du terminal X.

Une station de travail X fait tourner un serveur X, mais elle fait aussi tourner la plupart des processus sur son propre processeur et a le plus souvent ses propres disques locaux. Au contraire d'un terminal X qui doit nécessairement être connecté à un ordinateur par un réseau, une station de travail peut donc fonctionner en unité autonome bien qu'elle soit conçue pour être intégrée dans un réseau.

L'avantage des terminaux X est que le parc de postes de travail est plus facile à gérer que pour des stations de travail car tous les logiciels et les fichiers de données sont concentrés sur quelques ordinateurs. Ils sont aussi moins chers que les stations de travail. Leur inconvénient est qu'ils chargent davantage le réseau que les stations car ils génèrent beaucoup d'échanges avec le serveur X. D'autre part ils sont moins bien adaptés que les stations de travail pour faire fonctionner les applications gourmandes en temps processeur. En effet, dans ces cas-là les processeurs des serveurs de logiciels ou les réseaux sont rapidement saturés.

1. les nouvelles générations de terminaux X offrent la possibilité de lancer localement sur le terminal X des utilitaires ou un gestionnaire de fenêtres

Chapitre 7

Réseau Archipel de l'université

7.1 Description du réseau

Le réseau de l'université de Nice Sophia-Antipolis utilise pleinement les techniques d l'informatique répartie. Le nom du réseau est *Archipel*.

Il est constitué de réseaux locaux Ethernet sur différents campus de Nice et de Sophia Antipolis. Ces réseaux locaux sont interconnectés par des lignes fournies par les service de France Télécom. Plus de 1000 machines sont connectées au réseau Archipel.

Le réseau Archipel est géré par le Centre de Ressources Informatiques (CRI) de Valros

7.2 Réseau local de Valrose

NFS est utilisé pour fournir d'une façon transparente un environnement logiciel et u système de fichiers qui est réparti sur plusieurs machines.

Voici quelques serveurs :

taloa (adresse IP 134.59.1.7) est serveur de messagerie et de news, serveur de noms pri cipal pour les applications TCP/IP.

samoa (adresse IP 134.59.1.1) Le système d'exploitation est un Unix OSF alors que l autres machines Unix ont pour la plupart des Unix de type Système V (Solaris). L utilisateurs qui ont des comptes sur plusieurs machines pourront remarquer quelqu incompatibilités entre les deux systèmes. Quelques terminaux X ont des problèm pour se connecter directement à samoa.

clio, thalie et erato (adresses IP 134.59.2.190, 134.59.2.167 et 134.59.2.51) sont des se veurs de fichiers et de logiciels pour les étudiants et les enseignants du départemen d'informatique.

7.3 Connexions avec l'extérieur

7.3.1 Réseaux régionaux ou nationaux

Le réseau Archipel est relié par France Télécom au Réseau Régional de Télécommunication Recherche et Technologie (R3T2), au Réseau National de Télécommunication et de la Recherche (RENATER) et au réseau TRANSPAC. Le réseau RENATER relie les universités françaises.

7.3.2 Réseaux internationaux

Archipel est relié à Internet par l'intermédiaire de RENATER.

Le nom Internet du réseau Archipel est *fnet-unice*; son nom de domaine Internet est *unice.fr*. C'est un réseau de classe B, d'adresse IP 134.59.

7.3.3 Accès à Archipel via un minitel ou un modem

Toute personne autorisée peut accéder à Archipel par un modem ou un minitel. Il faut pour cela obtenir un compte spécial auprès du CRI de Valrose (très rarement attribué aux étudiants car les lignes sont peu nombreuses).

Connexion par un modem

L'accès par modem peut se faire par le réseau commuté (identique à un appel téléphonique) ou par TRANSPAC. On arrive sur une passerelle. Les commandes disponibles s'obtiennent en tapant la commande "help".

Lorsque l'on se connecte au réseau Archipel avec un modem à partir d'un ordinateur compatible PC ou d'un Macintosh, on doit lancer sur l'ordinateur personnel un logiciel grâce auquel l'ordinateur personnel joue le rôle d'un simple terminal alphanumérique connecté sur une des machines du réseau.

Pour transférer des données du système distant sur l'ordinateur personnel, ce type de logiciel offre souvent le copier/coller. Ce procédé ne fonctionne que pour les petits transferts de données. Pour le transfert de gros fichiers le copier/coller occasionne souvent des pertes de données. Il est donc préférable d'utiliser des logiciels de transfert comme *kermit*. kermit est lancé à la fois sur le système distant et sur l'ordinateur personnel. Les deux processus kermit entrent en contact et commencent le transfert des données en contrôlant la bonne arrivée des données. Si une erreur est détectée, la portion de fichier qui a été transférée avec erreur est à nouveau transmise.

Une passerelle "SLIP" et "ppp" (deux protocoles pour se connecter à Internet avec un modem) permet de connecter un ordinateur personnel à Internet par modem comme si l'ordinateur était directement branché sur le réseau Ethernet de l'université.

Connexion par un minitel

Pour se connecter par minitel, il faut tout d'abord passer le minitel en mode 80 co
lonnes en tapant "Fnct-T A" avant de composer le numéro de téléphone fourni par le CR
Quand on est connecté, on peut effacer le dernier caractère tapé en tapant [Ctrl] H (
pourra être nécessaire de lancer la commande Unix "`stty erase ^H`" pour que [Ctrl]
soit bien pris comme caractère d'effacement). On utilise la touche surmontée de l'indic
tion "E.page" située en bas à droite du clavier des minitels, comme touche "Return" pou
entrer les commandes (Ctrl J fonctionne aussi). Si une entrée "minitel" existe dans le fichi
`/etc/termcap` (ou un fichier terminfo en Unix système V), il est préférable de donner
valeur "minitel" à la variable d'environnement Unix `TERM`, sinon la valeur "vt100" convien

Index

.Z, 28
.gz, 29
.netrc, 23
.newsrc, 40
.plan, 30
.project, 30
.rhosts, 18
.tar, 29
/etc/hosts, 13, 17
/etc/hosts.equiv, 18
/etc/services, 14
10baseT, 8

activité réseau, 15
adresse Internet, 17
adresse IP, 12
Archie, 34
archie, 35
Archipel, 48
archive shell, 29

Binhex, 30

classe d'un réseau IP, 12
client-serveur, 16
commutation de circuit, 2
commutation de messages, 2
commutation par paquets, 2
compress, 28
connexion sur une machine distante, 19
copier des fichiers distants, 22
couches OSI, 4
courrier électronique, 32
CRI, 48
CSMA/CD, 8
CTAN, 25, 27

daemon, 16
datagramme, 2
datagramme IP, 11
DNS, 12, 17

emacs, 29, 32
Ethernet, 8
exécuter une commande distante, 20

FAQ, 39
finger, 30
ftp, 23
ftp anonymous, 26
full-duplex, 13

GNU, 27
Gopher, 35
gopher, 36
groupes du réseau, 17
gzip, 29

html, 37
http, 38
hypertexte, 37

ICMP, 14
informations sur le réseau, 30
Internet, 10
IP, 10, 11
IRC, 33
irc, 33
ISO, 4

kermit, 30, 49

LAN, 1
LaTeX, 25
ligne point à point, 1

liste de diffusion, 40
liste de messagerie, 40

machines du réseau, 31
mail, 29, 32
make, 29
mesg, 33
mftp, 29
mh, 32
minitel, 50
modem, 49

netstat, 15
News, 38
NFS, 45
NIC, 10, 12
NIS, 17, 46

OSI, 4

pages jaunes, 46
paires torsadées, 8
passerelle, 13
ping, 31
pont, 9
port TCP, 14
ppp, 49
protocole, 4

R3T2, 49
répéteur, 9
rcp, 22
récupérer des sites Web, 44
RENATER, 49
réseau étendu, 1
réseau de transmission, 2
réseau local, 1
réseaux à diffusion, 1
RFS, 45
rlogin, 19
routage, 5, 11
routeur, 11, 13
rsh, 20, 31
ruptime, 31
rwho, 31
rwhod, 31

serveur de noms, 12
shar, 29
SLIP, 49
SMTP, 32
sous-réseau, 12
station de travail, 46
swais, 37

talk, 33
tar, 29
TCP, 10, 13
TCP/IP, 10, 16
telnet, 20
terminal X, 46
topologie, 3
traceroute, 31
trame Ethernet, 8
TRANSPAC, 3, 49
type d'un réseau, 1

UDP, 10, 14, 33
uncompress, 28
uptime, 31
URL, 38
USENET, 38
uucp, 39
uudecode, 30
uuencode, 30

Veronica, 35

W3, 37
Wais, 36
WAN, 1
WWW, 37

X Window, 46
X.21, 2
X.25, 3
xarchie, 35
xftp, 29
xgopher, 36

xmh, 33
xrn, 39
xtalk, 33
xwais, 37

Yellow Pages, 46
ypcat, 17
ypmatch, 17

zcat, 28, 29