CONCEPTION DU SOUS-SYSTÈME RÉSEAU DES SYSTÈMES D'EXPLOITATION : LES CAS LINUX VOLUME UN ETHERNET, IP et UDP

Patrick Cegielski
cegielski@univ-paris12.fr

Juin 2005

Pour Irène et Marie

Legal Notice

Université Paris 12 - IUT
Route forestière Hurtault
F-77300 Fontainebleau
cegielski@univ-paris12.fr

Préface

On peut distinguer quatre niveaux en ce qui concerne les rapports entre une personne et le sous-système réseau d'un système d'exploitation :

- Au *niveau utilisateur*, on utilise des applications (logiciels) faisant intervenir une connexion informatique, telles que le courrier électronique, la consultation de pages web, le transfert de fichiers.
- Le *niveau administrateur* consiste à paramétrer le sous-système réseau du système d'exploitation, à le tenir à jour et à apporter des ressources aux utilisateurs (attribution d'une adresse IP, par exemple).
- Le *niveau programmation réseau* permet de concevoir des applications, que celles-ci soient utilisées localement ou connaissent une grande diffusion. Cette programmation se fait presque toujours en langage C en utilisant des *appels système* constituant une API (*Application Programmer Interface*).
- Le *niveau conception du noyau réseau* consiste à concevoir et à implémenter cette API réseau.

Le but de ce livre est de faire comprendre l'implémentation du sous-système réseau des systèmes d'exploitation, c'est-à-dire de l'API réseau. Il existe de très nombreux livres (en nombre trop important même) pour décrire le niveau utilisateur, qui fait de toute façon maintenant partie de la culture de base de tout un chacun. Il est même partie intégrante du "socle fondamental" de l'école tel que défini en France en 2004. L'administration réseau a également donnée lieu à une littérature abondante, par exemple [K-D–01]. La littérature sur la programmation réseau est moins fournie : la référence est [STE–90] dans le cas Unix. La littérature sur l'implémentation du sous-sytème réseau est plus que maigrichonne ([HER–00], [WPRMB–02], [C-P–02]). Il n'existe aucun livre général à notre connaissance.

Le choix du système d'exploitation permettant d'illustrer notre propos s'est tout naturellement porté sur Linux, puisque nous pouvons nous en procurer les sources très facilement.

Notre but n'est pas d'étudier TOUTE l'implémentation du sous-système réseau (Linux), ce qui occuperait plusieurs volumes comme celui-ci, dont le nombre de pages est déjà suffisamment imposant. Nous voulons simplement, si l'on peut dire, étudier un cas réel depuis le début jusqu'à la fin.

Dans la première partie, nous donnons une vue d'ensemble sur les réseaux et les modèles d'architecture (OSI à sept couches, TCP/IP à trois couches et le modèle mixte à cinq couches), nous décrivons très rapidement la couche physique (première couche) puisque la programmation (et donc le sous-système réseau) n'y a que faire. Pour illustrer la couche liaison (deuxième couche), nous avons choisi Ethernet, qui est la référence pour Linux. La couche réseau (troisième couche) est illustrée avec IP, devenu la référence absolue quelle que soit le réseau. Bien que le modèle de référence s'appelle TCP/IP, nous avons choisi d'illustrer la couche transport (la quatrième

couche) avec UDP et non TCP, ce qui est plus simple bien que déjà suffisamment complexe comme cela. Il n'y a pas à s'interroger sur le choix de l'API réseau puisque l'API des sockets est, comme IP, devenu la référence presque exclusive. Lors de la présentation de cette API, nous donnons quelques programmes très simples, ce qui nous aura permis de décrire le sous-système réseau d'un bout à l'autre.

La première partie reste générale, l'implémentation Linux n'y joue aucun rôle. Dans la seconde partie, nous étudions l'implémentation Linux de la réception d'un datagramme UDP, depuis la trame Ethernet et en passant par le paquet IP ordinaire. Nous commençons par la réception, même s'il a bien fallu l'envoyer, car son implémentation est plus simple que l'envoi.

La troisième partie traite de l'envoi d'un datagramme UDP.

La figure 0-1 montre la hiérarchie entre les chapitres.

	Réception	Général	Linux	Envoi
Généralités		ch1	ch10	
TCP/IP		ch5 ch9	ch11 ch12 ch13	
Couche physique		ch2		
Couche Liaison	 ch19	ch3 ch4 ch8	ch14 ch15 ch16 ch17	 ch34
Couche réseau	 ch23 ch36		ch18 ch20 ch21 ch22	 ch33 ch36
Couche transport	 ch25		ch24	 ch32
API réseau	 ch30	ch6 ch7	ch26 ch27 ch28 ch29 ch35	 ch31

FIG. 1 – *Hiérarchie entre les chapitres*

Le sous-système réseau est devenu partie intégrante des systèmes d'exploitation actuels. Nous supposons donc que le lecteur ait quelques notions sur l'implémentation des systèmes d'exploitation en général et de Linux en particulier. Il existe une littérature assez conséquente sur ce sujet, souvent excellente. Nous nous permettons de recommander notre livre [CEG–03], qui illustre les propos par le tout premier noyau Linux, qui ne concerne donc que l'essentiel, ce qui est suffisant pour aborder ce volume.

Ce livre n'a d'autre but que de publier en un seul volume les aspects suivants de l'implémentation d'un sous-système réseau :

- les concepts généraux sous-jacents à l'implémentation d'un sous-système réseau ;
- les concepts fondamentaux d'Ethernet et de TCP/IP ;
- la documentation sur un contrôleur de périphérique réseau, à savoir le carte 3Com 501 pour compatible PC ;
- une présentation des choix faits pour l'implémentation de Linux, suivie d'extraits de fichiers sources, repérables facilement par l'indication Code Linux située en marge, paraphrasés en français ; ces paraphrases ne sont pas théoriquement indispensables mais oh combien appréciables en pratique.

Nous parlerons quelquefois de "fonction générale" en décrivant rapidement le comportement de celle-ci mais sans nous reporter aux sources dans le cas où cela ne concerne pas les réseaux à proprement parler. Cet ouvrage est déjà suffisamment imposant comme cela par sa taille pour que nous n'ayons pas à revoir toute l'implémentation du système d'exploitation.

L'index est une partie fondamentale du livre, et pas seulement par le volume occupé (à peu près 30 pages) :

- Il s'agit tout d'abord d'un index des concepts fondamentaux concernant les réseaux. On s'y réfèrera pour chercher rapidement un mot dont on ne connaît pas la signification ou pour lequel on voudrait plus d'informations. Il renvoie à la page où il est défini.
- Il s'agit également d'un dictionnaire français-anglais et anglais-français de ces mêmes concepts. La dénomination anglaise de tout concept est toujours indiquée entre parenthèse lorsque celui-ci est introduit (en français). Cet index servira certainement également à ceux qui, soucieux de la langue française, veulement éviter les anglicismes. Pour tel mot anglais rencontré, l'index indique à quel page le mot français correspondant (et sa traduction anglaise entre parenthèses) est défini.
- Il s'agit également d'un index des grands noms qui ont participé aux réseaux, bien que cet aspect n'a pas encore été assez développé à notre goût (peut-être pour une édition ultérieure?).
- Les sources, de façon générale et de Linux en particulier, sont souvent d'approche un peu repoussante car il est difficile de s'y retrouver. On dispose maintenant pour cela de l'excellent outil `lxr` que nous décrivons au chapitre 10. Celui-ci connaît cependant quelques limites (il n'indexe pas les constantes définies dans un type énuméré par exemple) et, surtout, il ne renvoie pas aux commentaires adéquats. Notre index référence **toutes** les constantes, macros, variables, fonctions et tous les types fondamentaux de l'implémentation du sous-système réseau de Linux, renvoyant à la page où la définition est citée et commentée. La version du noyau choisie pour cette édition est 2.6.10, c'est-à-dire la dernière version au moment de la fin de la rédaction de ce livre.
- On n'a certainement pas intérêt à étudier les sources fichier par fichier car la philosophie de mise en fichier n'est pas suffisamment cohérente pour Linux. On peut cependant, à l'occasion de l'étude de tel ou tel point d'un fichier, se demander si cette partie du fichier est commentée dans ce livre. C'est pourquoi les fichiers sont indexés, avec les lignes des fichiers comme second niveau d'indexation.

Notre espoir serait, tel l'aigle, de pouvoir survoler cet immense champ qu'est le réseau pour avoir une vue d'ensemble et, là où ça nous intéresse plus particulièrement, piquer jusqu'au sol pour en voir le moindre détail.

Cet espoir est-il réalisé? C'est autre histoire mais c'est en tous cas le but vers lequel nous devons tendre.

Table des matières

Table des figures

Chapitre 1

Les réseaux informatiques

1.1 Introduction aux réseaux

1.1.1 Origine des réseaux informatique

La notion de calcul est l'une des grandes conquêtes de l'humanité, certainement apparue au néolithique pour les besoins de comptabilité des ressources indispensables pour les premières cités, concernant le cheptel et les réserves de nourriture. Les grandes quantités manipulées pour cette comptabilité a conduit à rechercher rapidement des outils d'aides au calcul (petits cailloux, puis abaque, puis machine arithmétique de Pascal...). La notion d'ordinateur, outil universel d'aide aux calculs dans la mesure où, par définition, un ordinateur est capable de calculer ce qui est calculable par n'importe quel outil, date de 1936 avec la définition par Alan TURING de sa célèbre machine (en fait un modèle, car il s'agit d'une machine imaginaire). La réalisation effective d'ordinateurs attendra encore quelques années, puisque le premier date de 1949[1].

Quelques années après, il fut envisagé de partager l'unité centrale d'un ordinateur entre plusieurs sites, soit pour des raisons de bonne économie, soit pour des raisons intrinsèquement liées à l'application envisagée (projet SAGE [pour *Semi-Automatic Ground Environment*, environnement au sol semi-automatique] de surveillance radar des États-Unis). L'alliance des télécommunications et de l'informatique était née.

Au milieu des années 1960 vint l'idée de relier les unités centrales elles-mêmes.

1. Ce que l'on vend sous le nom d'ordinateur n'est pas réellement une machine universelle. Mathématiquement ce sont des automates finis à un très grand nombre d'état et non des machines de Turing.

1.1.2 Notion de réseau informatique

Un **réseau informatiques**[2] (*computer network* en anglais) est un ensemble d'ordinateurs et de périphériques (imprimantes, traceurs, scanners, etc.) connectés ensemble par le biais d'un support physique (en anglais *medium*). La connexion peut être directe (par câble coaxial, par exemple) ou indirecte (par modem).

1.1.3 Matériel et logiciel pour réseau

Comme pour les systèmes informatiques[3], la distinction entre *matériel* et *logiciel* est importante dans le cas des réseaux.

L'objet de ce livre concerne le logiciel et non le matériel. Le matériel réseau nous apparaîtra comme une boîte noire fonctionnant parfaitement. Nous ne dirons du matériel que ce qui en est strictement nécessaire pour comprendre le logiciel associé.

1.1.4 Protocoles

1.1.4.1 Notion de protocole

Dans un réseau informatique, lorsqu'un ordinateur envoie des informations à un autre, les matériels et les logiciels sont en général différents. On a donc besoin d'un ensemble de règles pour coordonner l'échange de ces informations. Celles-ci forment un **protocole**, nom donné par Tom Marill en 1966 ([HL–96], p.83).

Le mot provient évidemment du langage diplomatique. Les diplomates suivent des règles lors des discussions entre nations, appelées un protocole. Le protocole diplomatique indique qu'il ne faut pas insulter ses hôtes et qu'il faut tâcher de respecter les coutumes locales. La plupart des ambassades et des consulats abritent des spécialistes du protocole, dont le rôle est de s'assurer que tout se passe harmonieusement lors des rencontres. Le protocole est un ensemble de règles qui doivent être suivies pour "jouer le jeu", pour reprendre une expression diplomatique.

1.1.4.2 Suite de protocoles

Les protocoles ont évolué. De processus très simples ("je t'envoie un caractère, tu me le renvoies, et je m'assure que les deux correspondent") au départ, ils sont devenus des mécanismes complexes qui prennent en compte tous les problèmes et conditions de transfert possibles. Un protocole unique qui couvrirait tous les aspects du transfert serait de taille trop importante, difficile d'emploi et trop spécialisé. Plusieurs protocoles ont donc été développés, chacun s'acquittant d'une tâche spécifique.

On appelle **suite de protocoles** un ensemble de protocoles cohérents qui couvre pratiquement tous les besoins de communication.

1.1.5 Systèmes propriétaires et systèmes ouverts

Au début, il n'y avait qu'un réseau (ARPANET) avec ses protocoles associés. Devant le succès de celui-ci, des produits commerciaux furent développés, plus particulièrement pour les réseaux

2. Désormais nous ne parlerons que de *réseau* au lieu de *réseau informatique*. L'histoire des premiers réseaux de télécommunication (premiers essais, télégraphe optique, télégraphe électrique, télégraphie d'images, téléphone, transmission radio et systèmes de commutation) est contée dans [HUU–03].

3. On appelle **système informatique** un poste de travail informatique complet : ordinateur, bien sûr, mais aussi périphériques (tels que imprimante ou scanner) et logiciels. De nos jours, les outils réseau font partie intégrante du système informatique, dans un sens plus large.

locaux et étendus. Chaque constructeur avait sa ligne de produits, matériels comme logiciels. On parle de **système propriétaire**. Les sources, et même les spécifications complètes, des systèmes propriétaires sont rarement diffusées. Ceci présente une difficulté pour la conception des inter-réseaux.

Par opposition, un **système ouvert** est un système dont au moins les spécifications complètes sont diffusées, éventuellement les sources des logiciels.

Le mouvement des systèmes ouverts n'est pas né à propos des réseaux, même si c'est là qu'il a connu son plus grand succès puisqu'il n'existe plus de système propriétaire, mais à propos des systèmes d'exploitation. Jusqu'aux années 1980, chaque constructeur de matériel avait sa ligne de produits, et on était dans une très large mesure dépendant de ce constructeur pour tout ce qui concernait le logiciel et le matériel. Certains constructeurs profitaient de cette situation pour pratiquer des tarifs prohibitifs ou pour imposer certaines configurations à leurs clients. Le ressentiment des clients prit une telle ampleur que ces derniers finirent par imposer leurs souhaits. IBM commença par diffuser le code du BIOS de ses micro-ordinateurs. DEC (*Digital Equipment Corporation*) passa d'un système d'exploitation propriétaire, VMS, sur ses mini-ordinateurs à un système d'exploitation ouvert de type UNIX. Ses clients lui en furent gré et l'entreprise vendit plus de machines. Microsoft essaie de faire passer sa technologie .NET comme un standard ECMA (*European Computer Manufacturer Association*), sans totalement y réussir pour l'instant (seule la spécification du langage C# est définie par un standard ECMA pour l'instant, début 2005).

1.2 Réseau et commutation

Les éléments d'un réseau de télécommunication doivent être reliés entre eux et ceci sur une distance quelquefois très éloignée. Comment réaliser ceci?

1.2.1 Notion de commutation

En 1876, peu de temps après qu'Alexander Graham Bell eut déposé son brevet d'invention du téléphone, il y eut une vague énorme de demandes. Au début, les téléphones étaient vendus par paire et il appartenait au client de les relier à l'aide d'un fil électrique (la voie du retour se faisait par la terre). Si une personne souhaitait converser avec n autres interlocuteurs, il lui fallait être raccordé par autant de fils aux n domiciles de ces derniers. On aboutit à ce qu'on appelle un graphe complet, représenté à la figure 1.1.

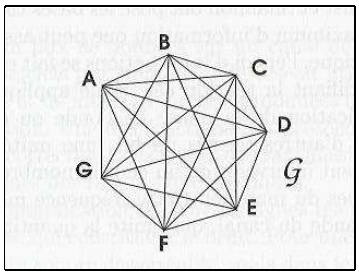


FIG. 1.1 – *Graphe complet*

En l'espace d'un an les villes se trouvèrent prises dans un enchevêtrement sauvage de fils passant par-dessus les toits et les arbres. Il devint très vite clair qu'un modèle d'interconnexion où chaque téléphone était relié à tous les autres n'était pas viable.

Bell réagit à cette situation en fondant la société Bell Telephone Company, qui créa la notion de **central téléphonique** et mit le premier central en place à New Heaven (Connecticut) en 1878. La société amena un fil électrique à chaque domicile ou bureau d'*abonné*. Pour effectuer un appel, le client devait tourner une manivelle qui produisait une sonnerie au niveau du central téléphonique, attirant l'attention d'une opératrice. Celle-ci se chargeait ensuite de raccorder manuellement la ligne de l'appelant à celle du correspondant appelé au moyen d'un câble de jonction sur un *tableau de commutation*. Ce modèle centralisé est illustré à la figure 1.2.

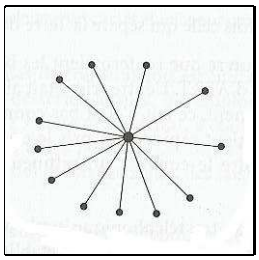


FIG. 1.2 – *Réseau avec commutateur central*

Très rapidement d'autres centraux furent créés un peu partout. Et comme les clients souhaitaient pouvoir appeler des correspondants dans d'autres villes, il a fallu interconnecter ces centraux. Ce fut, là encore, d'abord un graphe complet, puis des centraux de second niveau et ainsi de suite. Il y eut jusqu'à cinq niveaux de centraux.

1.2.2 Types de commutation

Les divers réseaux de télécommunication ont conduit à plusieurs types de commutation, comme nous allons le voir.

1.2.2.1 Commutation de circuits

Si d'un poste téléphonique de New-York vous demandez le 22 à Asnières à une opératrice, celle-ci contacte le central interurbain, qui va contacter le central reliant le câble transatlantique à un central en Europe, qui contacte un central en France, qui contacte un central dans la région parisienne, qui contacte le numéro demandé. Par un jeu de fiches dans chacun de ces six centraux, un chemin physique est établi de l'appelant vers l'appelé. Un tel chemin est appelé un **circuit** dans le jargon de la téléphonie (même s'il ne s'agit pas d'un circuit au sens de la théorie des graphes). On parlera plus tard de **commutation de circuits** (*circuit switching* en anglais) pour ce type de commutation lorsque d'autres types de commutation apparaîtront. Une ligne physique est dédiée à cette communication durant tout le temps de la communication, d'où le paiement à la durée de la communication.

Au début de l'ère de la téléphonie, la connexion physique était établie manuellement comme nous venons de le rappeler. On parlera plus tard de **commutation manuelle**. Peu après l'invention du téléphone, un équipement automatique de commutation électro-mécanique fut inventé par Almon Strowger et fut utilisé pendant près de cent ans. Au début des années 1970, il fut remplacé par un équipement électronique, mais le principe est le même.

Le gros avantage de la communication de circuits est qu'une ligne est entièrement dédiée à la communication. Elle présente cependant également trois gros inconvénients :

– Le temps d'établissement d'une ligne, en passant à travers tous les centraux, n'est pas

négligeable. Ce qui est acceptable pour une communication téléphonique le serait moins pour un réseau informatique.
- Il peut arriver que toutes les liaisons d'un central à l'autre soient occupées. On vous demande alors de renouveler votre appel ultérieurement.
- Le débit n'est pas utilisé de façon optimale. Lors des silences dans la communication téléphonique, qui peuvent parfois être longs lorsqu'un des correspondants recherche quelque chose, le circuit est toujours dédié à cette communication.

1.2.2.2 Commutation de messages

Une autre technique de commutation est la **commutation de messages** (*message switching* en anglais). Lorsqu'elle est mise en œuvre, aucun chemin physique n'est établi au préalable entre l'émetteur et le récepteur. Lorsque l'émetteur envoie un bloc de données, celui-ci est stocké dans le premier central de commutation (appelé **routeur** dans ce cas), contrôlé pour vérifier qu'il ne comporte pas d'erreurs puis transmis au routeur suivant. On parle de **transmission différée** (*store-and-forward* en anglais).

Cette technique de commutation fut mise en œuvre dans les premiers systèmes électromécaniques de télécommunications pour transmettre les télégrammes.

L'énorme avantage est que le débit est utilisé au mieux et que le temps d'établissement du circuit n'existe plus. Cependant les stockages successifs entraînent des délais qui empêcheraient l'utilisation d'une telle commutation pour la voix, par exemple.

1.2.2.3 Commutation par paquets

Dans la commutation de messages, la taille des blocs n'est pas limitée. Cela a pour conséquence que les routeurs doivent disposer de disques pour placer en mémoire tampon les longs blocs. Un seul bloc peut monopoliser une ligne entre deux routeurs pendant des minutes. Cette technique de commutation est donc inadaptée au trafic interactif. De plus, si un gros message a été altéré, il est totalement perdu ou il faut le renvoyer intégralement.C'est pourquoi la **commutation par paquets** (*packet switching* en anglais) a été developpée.

La commutation de paquets est analogue à la commutation de messages mais les messages sont découpés en paquets d'une taille maximale, ce qui évite les disques tampon et la congestion durant plusieurs minutes due à un gros message.

Les paquets empruntent éventuellement des chemins différents, ce qui peut entraîner des arrivées désordonnées. On numérote donc les paquets.

Le premier paquet d'un message qui en comporte plusieurs peut être transmis par un routeur avant que le deuxième paquet ne soit complètement arrivé, ce qui réduit le délai et améliore le débit.

La commutation de paquets offre une meilleure tolérance aux pannes que la commutation de circuits. Si un routeur devient indisponible, les paquets peuvent le contourner et poursuivre leur chemin.

1.2.2.4 Cas des réseaux informatiques

Dans le cas des réseaux informatique, on utilise le plus souvent la commutation de paquets (c'est le cas pour Internet) et plus rarement la commutation de circuits (c'est le cas de ATM), mais jamais la commutation de messages.

On appelle **routage** l'algorithme utilisé par un routeur pour déterminer l'ordinateur ou le routeur auquel il faut envoyer le paquet qu'il vient de recevoir pour que celui-ci arrive à destination au vu de l'adresse réseau.

1.2.3 Réseaux hiérarchisés et réseaux maillés

Les réseaux informatiques se distinguent des autres réseaux de télécommunication dans la mesure où ils sont maillés et non hiérarchisés. L'utilisation de réseaux distribués au lieu des réseaux hiérarchisés utilisés en téléphonie fut prônée par Paul Baran avant de devenir réalité lors de la réalisation des réseaux informatiques.

Paul Baran s'intéressa dès 1960 à la capacité de survie des systèmes de communication en cas d'attaque nucléaire. À l'époque, les réseaux de communication à longue distance étaient extrèmement vulnérables et hors d'état de supporter une attaque nucléaire. Pourtant le pouvoir qu'avait le président des États-Unis de commander ou d'annuler l'envoi de missiles nucléaires reposait sur ces systèmes de communications vulnérables. Baran a été l'un des premiers à établir, au moins de façon théorique, que le problème pouvait être résolu. Il était arrivé à l'idée qu'un réseau de transmission de données pouvait être rendu plus robuste et plus fiable en introduisant des niveaux de redondance élévés. Le mécanisme de défense qui consiste à diviser une vaste structure, unique et vulnérable, en de nombreuses parties, se retrouve dans maintes applications, comme le compartimentage de sécurité des navires. Théoriquement, il était possible d'installer un réseau avec de nombreuses connexions redondantes. Mais il y avait une restriction technique : tous les signaux sur le réseau téléphonique étaient analogiques. Le plan d'acheminement sur ce réseau interdisait d'utiliser plus de cinq liaisons les unes à la suite des autres à cause de l'altération du signal. Baran proposa donc un agencement en **réseau réparti** (dit aussi **distribué**) comme le montre la figure 1.3 [BAR–60, BAR–64].

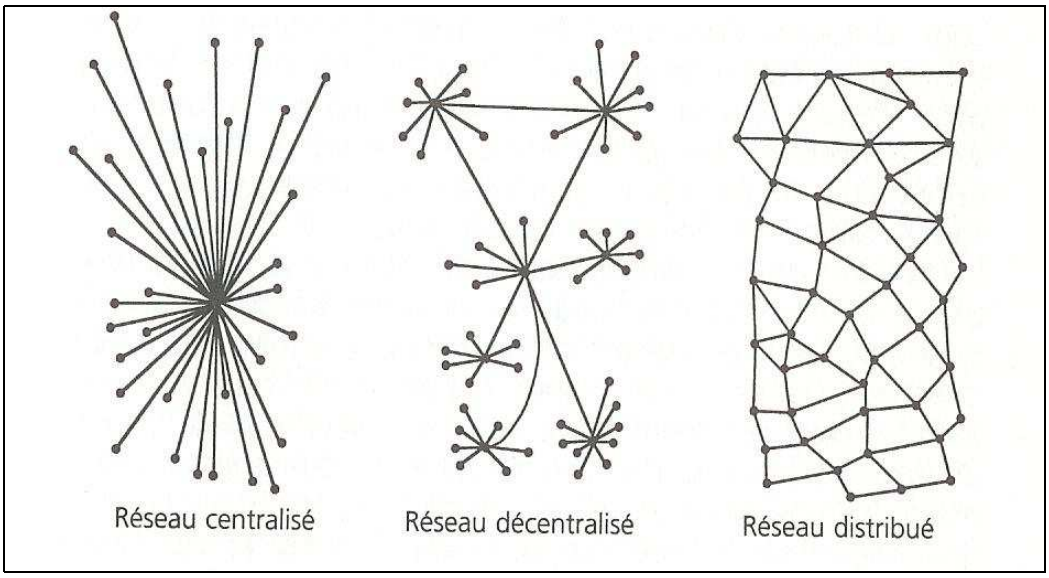


FIG. 1.3 – *Réseau distribué*

Néanmoins une question demeurait, nous faisant passer du qualitatif au quantitatif : quel degré de redondance fallait-il introduire dans les connexions entre nœuds voisins pour garantir la persistance du réseau? Baran effectua de nombreuses simulations. Il en conclut qu'il suffirait d'un faible niveau de redondance : que chaque nœud soit connecté à trois ou quatre autres permet d'offrir un niveau exceptionnelement élévé de résistance et de fiabilité (voir [HL–96], pp.64–72).

1.3 Modèles en couches

La partie logicielle des réseaux comprend un grand nombre de fonctions, chacune relevant d'un protocole. Un ensemble de protocoles cohérent couvrant l'ensemble des besoins pour un réseau s'appelle une **suite** de protocoles. L'ensemble d'une telle suite s'appelle un **modèle réseau** ou **architecture réseau**.

1.3.1 Étude générale

1.3.1.1 Notion

Imaginez que vous deviez écrire un programme qui fournisse des fonctions de réseau à toutes les machines de votre réseau local. L'écriture d'un logiciel unique se chargeant de toutes les tâches nécessaires à la communication entre plusieurs ordinateurs serait un vrai cauchemar. De plus, dans l'hypothèse où vous arriveriez à gérer tous les matériels présents sur le réseau, le programme résultant serait bien trop grand pour être maintenu ou même exécuté.

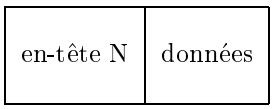
Il est plus raisonnable de diviser chaque domaine d'opérations en groupes de natures proches. Les groupes sont assez évidents à discerner. Un groupe traite du transport des données, un autre de l'empaquetage des messages, un autre des applications utilisateur, etc. Chaque groupe de tâches proches est appelé une **couche**. On obtient ainsi un **modèle en couches**.

Les couches d'une architecture réseau sont censées être des entités autonomes et indépendantes. Une couche ne peut évidemment pas effectuer une tâche observable sans interagir avec d'autres couches mais, du point de vue de la programmation, elles sont indépendantes. Elles ne doivent interagir les unes avec les autres que grâce à leur **interface** proprement définie.

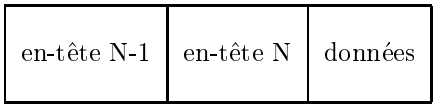
1.3.1.2 Pile réseau

Dans un modèle en couches, les couches sont numérotées de 1 à N, allant du niveau le plus proche du matériel (concernant le port série, puis la carte réseau) au niveau le plus proche de l'application de l'utilisateur (courrier électronique, transfert de fichier...). Plus on est proche du matériel, plus le numéro de couche est bas.

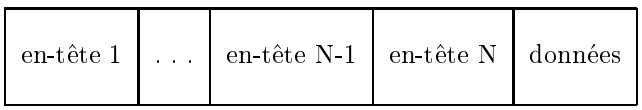
Chaque protocole encapsule les données dans un ensemble plus grand comprenant en général un **en-tête** et quelquefois un **suffixe**. Supposons qu'il n'y ait pas de suffixe. Les données de l'utilisateur sont encapsulées avec l'en-tête du niveau N :



Cela devient des données de niveau N qui sont encapsulées avec l'en-tête de niveau N-1 pour obtenir des données de niveau N-1 :



et ainsi de suite jusqu'aux données encapsulées de niveau 1 :



L'intérêt des modèles en couche est, qu'à chaque niveau, il n'est pas besoin de passer en revue l'ensemble des en-têtes mais uniquement celui du niveau correspondant.

On parle aussi de **pile réseau**, *stack* en anglais, car on peut considérer que les en-têtes des niveaux N à 1 sont empilés au-dessus des données de base lors de l'expédition et qu'elles sont dépilées lors de la réception, comme le montre la figure 1.3 ([K-R–01], p.52).

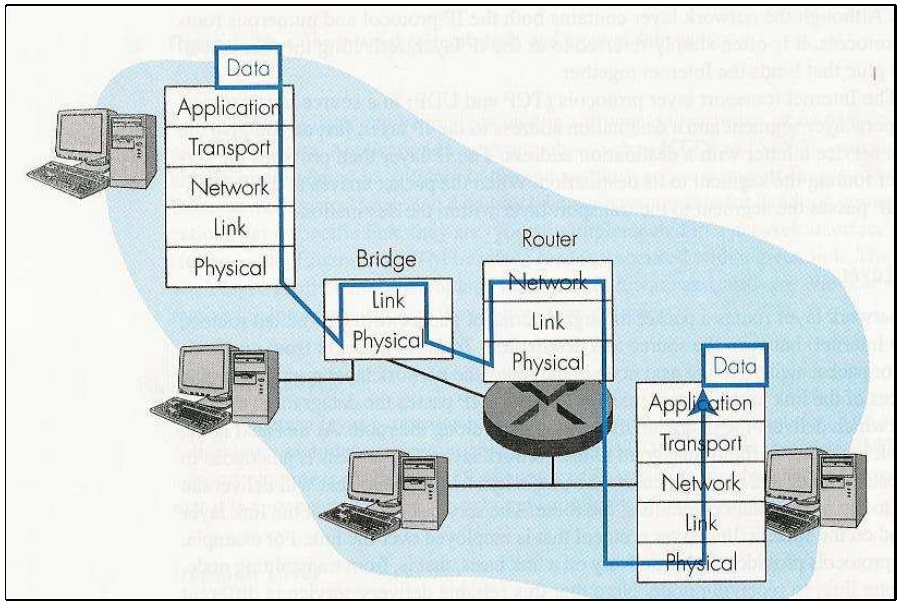


FIG. 1.4 – *Pile réseau*

1.3.1.3 Intérêts et réalisations

Les modèles en couches présentent beaucoup d'intérêt. La suite de protocoles est beaucoup plus simple à concevoir : le protocole d'un niveau donné peut être conçu par une équipe différente du protocole d'un autre niveau. Il en est de même de la suite de logiciels mettant en place ce modèle. De plus, les matériels actifs du réseau n'ont à considérer que les niveaux les plus bas, par exemple :

- un répéteur ne considérera que les données brutes, sans même entrer dans le détail, autrement dit ne concernera que la couche physqiue ;
- un routeur ne considèrera que les en-têtes des couches les plus basses pour obtenir l'adresse de destination.

Deux modèles furent développés pratiquement en parallèle : le **modèle OSI** (par l'organisme de standardisation international ISO) et la **suite TCP/IP** (d'après le nom de deux des protocoles de la suite). Rétrospectivement on s'aperçoit qu'un modèle est largement dominant, pour ne pas dire exclusif : la suite TCP/IP. Il est quand même intéressant de dire quelques mots du modèle OSI, même s'il est désormais pratiquement abandonné, car les deux couches les plus basses ne sont pas prises en considération par TCP/IP, ce qui a conduit à un modèle mixte.

1.3.2 Modèle OSI

L'ISO (*International Standardisation Organisation*), l'organisation de standardisation la plus prioritaire dans le monde entier, fondée en 1947, a proposé un modèle en sept couches en 1984 [ISO 7498-1], appelé **modèle OSI**, en fait **OSI–RM** pour *Open Systems Interconnection Reference Model* (modèle de référence d'interconnection des systèmes ouverts) :

7	Application	Couches supérieures
6	Signification	
5	Session	
4	Transport	Couches inférieures
3	Réseau	
2	Liaison des données	
1	Physique	

Le modèle OSI ne décrit aucune implémentation réelle d'un système particulier, mais se contente de définir les tâches des différentes couches.

Les couches d'application, de présentation et de session sont toutes les trois orientées application, c'est-à-dire qu'elles présentent l'interface de l'application à l'utilisateur. Ces trois couches sont totalement indépendantes des couches situées sous elles, elles ne connaissent rien de la façon dont les données parviennent à l'application. Elles sont appelées **couches supérieures**.

Les quatre **couches inférieures** se chargent de la transmission des données, en gérant l'empaquetage, le routage, la vérification et la transmission de chaque ensemble de données. Elles ne font aucune différence entre les diverses applications.

Détaillons maintenant le rôle de chacune des sept couches :

- La **couche physique** (*physical layer* en anglais) contrôle la transmission des différents bits *via* un support physique (*media* en anglais). C'est dans cette couche qu'on s'occupe de la façon dont les suites de bits sont converties (sans structuration) en signaux physiques et transmises *via* un support physique (câble de cuivre, fibre de verre, radio, etc.). La couche physique définit les procédures de codage physique (telle ou telle différence de potentiel par exemple), la géométrie des connecteurs enfichables et les types des supports spéciaux. Les protocoles de cette couche dépendent du support physique.
- La **couche de liaison des données** (*data link layer* en anglais) est chargée de la transmission correcte des données d'un point du réseau à un autre relié directement à lui *via* un support physique. Elle s'occupe de la correction des erreurs survenant lors de cette transmission (différentes des erreurs dans les données elles-mêmes, traitées dans la couche de transport). Elle doit tenir compte des interférences de signal (fréquentes, pouvant provenir de plusieurs sources, parmi lesquelles les rayons cosmiques et les interférences magnétiques provenant d'autres équipements).
- La **couche réseau** (*network layer* en anglais) est chargée de la transmission des données d'un point du réseau à un autre, que celui-ci soit relié directement à lui ou non. Il s'agit d'abord de déterminer un chemin (ou **route**, comme on préfère l'appeler dans le vocabulaire des réseaux) de l'expéditeur vers le destinataire en utilisant des machines intermédiaires : on parle du **routage** physique des données. Elle est chargée également de l'adaptation des unités de données à la taille admissible de la couche liaison existante : on parle de **fragmentation**.
- La **couche transport** (*transport layer* en anglais) se charge du multiplexage (lors de l'envoi) et du démultiplexage (lors de la réception), c'est-à-dire de distribuer aux différentes applications les paquets arrivés sans encombre.

Il y a **multiplexage** lorsque plusieurs communications transitent par un support physique unique. Le **démultiplexage** est l'inverse du multiplexage. Le multiplexage est indispensable pour prendre en charge de nombreuses connexions simultanément, tout en ne disposant que de ressources limitées. Un exemple classique est un bureau distant comprenant vingt terminaux. Chaque terminal pourrait être connecté au bureau principal par le biais d'une ligne téléphonique dédiée. Au lieu d'utiliser vingt lignes, on peut aussi s'arranger pour multiplexer les connexions afin de n'utiliser que trois ou quatre lignes téléphoniques.

Elle peut, de plus, assurer d'autres tâches. Elle peut établir, maintenir et terminer les communications entre deux machines (dans le cas des **communications** dites **connectées**). Elle peut se charge d'assurer que les données envoyées correspondent aux données reçues, tout au moins modulo certains points de comparaison et, si elles ne correspondent pas, demander à ce que les données soient envoyées à nouveau. Elle peut gèrer l'envoi des données, en déterminant l'ordre et la priorité de l'envoi.

- La **couche de session** (*session layer* en anglais) contrôle l'échange structuré de dialogues *via* les liaisons de communication. Il est, par exemple, possible de contrôler au cours d'une session si le transfert des données peut avoir lieu simultanément dans les deux sens ou si un seul partenaire à la fois dispose du droit d'émission. Dans ce dernier cas, la couche session gère le droit d'émission. Elle opère avec la couche d'application pour fournir des ensembles de données simples, appelés **points de synchronisation**, qui permettent à l'application de connaître l'état de progression de la transmission et de la réception des données. Il s'agit donc d'une couche de temporisation et de contrôle des flux.
- La **couche de présentation** (*presentation layer* en anglais) contrôle la présentation des données à transmettre sous une forme ne dépendant pas des systèmes. Elle convertit les données de l'application en un format commun, souvent appelé la **représentation canonique**, par exemple pour éviter le problème du code (Unicode, ASCII ou EBDIC) de représentation des caractères, le problème du format (petit-boutien ou grand-boutien) des entiers ou la façon d'indiquer le passage à la ligne.

  Le seul problème qui relève de cette couche que nous aborderons sera celui concernant le format (petit-boutien ou grand-boutien).
- La **couche d'application** (*application layer* en anglais) est l'interface utilisateur vers le système OSI. C'est là que résident les applications telles que le courrier électronique. La tâche de la couche d'application est d'afficher les informations reçues et d'envoyer aux couches inférieures les données fournies par l'utilisateur.

1.3.3 Suite TCP/IP

L'architecture TCP/IP est similaire au modèle OSI mais ne met en jeu que trois couches, car elle combine les couches supérieures OSI en une seule et ne s'occupe pas des couches en-dessous de la couche réseau, les protocoles des ces couches étant propres au réseau sous-jacent. Elle date de 1974 mais elle est définie, après coup en 1989, dans la section 1.1.3 de [RFC 1122] :

Application
Transport
Internet

- La **couche application** (*application layer* en anglais) regroupe toutes les tâches orientées application, c'est-à-dire celles des couches 5 à 7 du modèle OSI.
- La **couche transport** (*transport layer* en anglais), comme dans le modèle OSI, permet le multiplexage et le démultiplexage entre les applications de systèmes d'extrémité.

– La **couche Internet** (*Internet layer* en anglais) est principalement chargée de de router les paquets IP de l'expéditeur au destinataire à travers le réseau. Elle correspond à la couche réseau du modèle OSI.

1.3.4 Le modèle des sockets

Les modèles ci-dessus datent de 1984 et 1989. Tout se complique encore par le fait que les sockets, mises en place en 1983, comme nous le verrons plus loin, donnent lieu à un modèle du sous-système réseau des systèmes d'exploitation en trois couches avec un vocabulaire différent, mais que l'on peut relier aux modèles précédents :

– La couche physique n'interfère pas du tout avec le sous-système réseau des systèmes d'exploitation, donc on n'en parle pas.
– Les protocoles de la couche liaison sont traités dans la partie électronique (le contrôleur) des cartes réseau. Il n'y a donc pas besoin non plus d'en parler au niveau du sous-système réseau.
– La couche réseau s'appelle **famille d'adresses**, la mise en place des sockets retenant essentiellement la structure des adresses.
– La couche transport s'appelle **type de communication**.
– Une troisième couche, appelée **protocole**, avait été prévue si les deux couches précédentes n'étaient pas suffisantes. On la retrouve comme paramètre (presque toujours égal à zéro) mais ne sert pas à grand chose.
– Les couches supérieures concernent les applications et non le sous-système réseau, donc on ne s'en occupe pas.

1.3.5 Modèle hybride

L'Internet et la plupart des réseaux intranets d'entreprise ont recours à une **architecture hybride** TCP/IP–OSI qui s'appuie sur les couches basses de l'architecture OSI pour spécifier les infrastructures de réseau de type LAN ou autres :

Application
Transport
Réseau
Liaison
Physique

Nous verrons de plus que la couche liaison de données est divisée en deux sous-couches dans le standard IEEE.

1.3.6 Cas de Linux

Linux se réfère au modèle hybride mais la mise en place conduit à deux commentaires :

– Linux ne s'occupe pas du tout de la couche physique puisque aucun élément de programmation n'intervient à ce niveau. La mise en œuvre des protocoles de la couche liaison est presque entièrement traitée dans les cartes réseau de nos jours ; Linux n'a donc pas à s'en préoccuper si ce n'est sous la forme d'une interface avec celle-ci (il faut bien commencer par quelque chose).

– Linux ne respecte pas entièrement la philosophie des couches avec une interface bien définie. Nous verrons, par exemple, que l'on passe directemnt de la couche application à la couche réseau dans certains cas (plus exactement lors de l'envoi de données UDP).

1.3.7 Les en-têtes de protocole

Une unité d'information est composée de données et d'informations de contrôle de cette unité d'information. Ces informations de contrôle sont généralement assemblées dans un bloc venant avant les données, ce qui est le cas pour tous les protocoles de TCP/IP. On appelle ces informations un **en-tête de protocole**.

Lorsque l'unité d'information est passée à la couche inférieure, celle-ci ajoute son en-tête à l'unité d'information passée, considérée comme les données de cette couche inférieure. De cette manière, lorsqu'une unité d'information est partie de la couche d'application, au moment où il atteint la couche physique, elle contient quatre en-têtes de protocole (sept avec le modèle OSI).

Pour mieux visualiser ce processus, on peut se représenter les différentes couches d'un oignon. L'intérieur est constitué par les données à envoyer. À chaque fois que l'unité d'information passe à travers une des couches, une couche d'oignon est ajoutée. Lorsqu'elle a parcouru toutes les couches, plusieurs en-têtes de protocole entourent les données initiales. Lorsque l'unité d'information est envoyée à travers toutes les couches en partant du bas (sur une autre machine en général), chaque couche pèle l'en-tête de protocole lui correspondant. Lorsque la couche d'application de destination est atteinte, il ne reste plus que les données initiales.

OSI dispose d'une description formelle pour tout ce processus. La couche en cours porte le numéro N. Les données N-utilisateur à transférer doivent être précédées d'informations de contrôle de N-protocole (**N-PCI** pour *Protocol Control Information*) pour former une unité de données de N-protocole (**N-PDU** pour *Protocol Data Unit*). Les N-PDU sont passéss par l'intermédiaire d'un point d'accès de N-service (**N-SAP** pour *Service Access Point*) sous la forme d'un ensemble de paramètres de service comprenant une unité de données de N-service (**N-SDU** pour *Service Data Unit*). Les paramètres de service comprenant les N-SDU sont appelés les données d'utilisateur de N-service (**N-SUD** pour *Service User Data*), mis devant le (N-1)-PCI pour former un autre (N-1)-PDU.

1.4 Historique

Une bonne introduction à l'histoire des réseaux est [HL–96]. Elle a été écrite par des journalistes qui ont pu, d'une part, consulter les documents de littérature grise très difficilement accessibles et, d'autre part, interviewer les premiers acteurs de cette histoire. Elle contient une très grande documentation dont le seul reproche que l'on puisse faire est la non visibilité de la structuration par le manque de titres et sous-titres suffisamment explicites.

1.4.1 Naissance des ordinateurs

Il n'existe pas, à ma connaissance, d'étude abordable sur l'origine de la comptabilité au Néolithique, ni sur l'origine du calcul.

La notion de machine de calcul universel et l'apparition des premiers ordinateurs sont contés, à un niveau de haute vulgarisation avec des pointeurs sur la littérature primaire, dans [DAV–00], malheureusement non traduit en français.

1.4.2 Terminaux distants

Nous avons vu qu'une première alliance entre informatique et télécommunication est l'utilisation de terminaux distants de l'unité centrale, parfois situés à plusieurs milliers de kilomètres.

Le premier problème à résoudre pour cela est celui des utilisateurs multiples (*time-sharing* en anglais). Ceci est un problème du système d'exploitation. En 1961 est développé le **CTSS** (*Compatible Time Sharing System*) au MIT (*Massachussets Institute of Technology*), première réalisation importante mais encore expérimentale dans ce domaine.

Ceci conduira, en 1964, un groupe de chercheurs du MIT, des Bell Laboratories et de General Electric à s'associer pour initier le développement d'un système d'exploitation multi-utilisateur. Ils baptisèrent leur ambitieux projet **MULTICS** (*Multiplexed Information and Computing System*). Ce projet n'a pas vraiment abouti et sera abandonné en 1968. Il est cependant à l'origine de Unix, qui date de 1971.

Du point de vue matériel, dans les premiers essais d'utilisation des terminaux distants, on utilisait des lignes télex.

En 1974, IBM lance le **SNA** (*Systems Network Architecture*) qui normalise la communication entre un ordinateur et les terminaux distants. **VTAM** (*Virtual Telecommunication Access Method*) tourne sur l'ordinateur (un IBM 370) alors que **NCP** (*Network Control Program*) tourne sur le contrôleur de communication pour établir et surveiller en permanence le trafic.

1.4.3 Première mise en réseau : 1965

La première mise en réseau, entre deux ordinateurs (et non une unité centrale et un terminal), eut lieu en 1965. Le psychologue Tom Marill lança cette année-là une petite société de systèmes en temps partagé. Mais son principal investisseur ayant fait défaut à la dernière minute, il dut chercher un contrat de recherche et développement. Il proposa donc à l'ARPA de mener une expérience de mise en réseau, en joignant l'ordinateur TX-2 du Lincoln Laboratory et le SDC Q-32 situé à Santa Monica. La société de Marill était si petite que l'ARPA lui recommanda de procéder à son expérience sous l'égide du Lincoln Laboratory. L'idée plut aux responsables du Lincoln et ils chargèrent Larry Roberts de surveiller le projet.

La liaison entre les deux ordinateurs était réalisée grâce à un service spécial de la Western Union : quatre fils en duplex intégral. Marill branchait sur cette liaison un type de modem rudimentaire, opérant à 2 000 bits par seconde, qu'il appelait un composeur automatique (*automatic dialer*). Marill établit une procédure pour grouper les caractères en messages, les envoyer et vérifier qu'ils arrivent. Si aucun accusé de réception ne suivait, le message était transmis à nouveau. Il appela " protocole" de message l'ensemble des procédures pour faire circuler l'information dans les deux sens. En dépit de leurs efforts, lorsque Marill et Roberts connectèrent effectivement les deux machines, le résultat fut mitigé : le temps de réponse était médiocre (voir [HL–96], pp. 82–83).

Même si le résultat n'est pas vraiment celui escompté, il s'agit bien de la première mise en réseau : les ordinateurs ont des systèmes d'exploitation différents et on utilise des protocoles.

1.4.4 Réseaux de communication

De nos jours, "réseau" sans adjectif désigne toujours un réseau informatique. Le réseau est un concept général très utilisé depuis le réseau de nos connaissances personnelles jusqu'aux réseaux de communication et de télécommunication.

Les réseaux de communication cohérents commencent avec les Romains pour les routes, revus aux dix-neuvième siècle avec le réseau de routes nationales (départementales et vicinales) et celui des autoroutes au vingtième siècle sans oublier les chemins de fer.

Les réseaux de télécommunication commencent avec les feux dans l'Antiquité, les signaux de fumée des indiens d'Amérique, les relais de poste au début des Temps modernes. Il prit vraiment son essor avec le télégraphe optique de Chappe sous le Révolution puis avec le télégraphe électrique qui le détronera dans la seconde moitié du dix-neuvième siècle. C'est aussi le début des premiers câbles transatlantiques. Le réseau téléphonique suivra avec la naissance des grandes sociétés telles que AT&T aux États-Unis et les PTT dépendant directement du gouvernement dans de nombreux pays dont la France.

1.4.5 Commutation par paquets

L'utilisation de la commutation par paquets fut pronée indépendanmment par Paul Baran aux États-Unis et Donald Davies à Londes.

Nous avons vu ci-dessus comment Paul Baran en est venue à l'idée de l'utilisation des réseaux maillés. La seconde contribution de Baran aux réseaux a été plus révolutionnaire encore : disloquer également les messages pour obtenir la commutation par paquets, qu'il appelait " blocs-messages". Dans le modèle de Baran, chaque nœud de communication contenait une table de routage qui se comportait comme une sorte de coordinateur ou de régulateur. Il espérait persuader AT&T des avantages de son projet mais ce ne fut pas le cas. En 1965 cependant, cinq ans après s'être lancé dans le projet, Baran obtint l'appui complet de la RAND qui voulût construire un réseau de commutation distribué. Malheureusement AT&T, qui disposait du monopole des télécommunications, s'y opposa. Baran se tourna alors vers d'autres projets (voir [HL–96], pp.72–78).

À Londres, à l'automne 1965, juste après que Baran eut laissé tomber son projet, Donald Watts Davies, quarante et un ans, physicien au British National Physical Laboratory (NPL), écrivait une première note exposant ses idées à propos d'un nouveau réseau d'ordinateurs fort proche de celui de Baran. Le printemps suivant il donnait à Londres une conférence publique où il décrivait l'envoi de petits blocs de données – qu'il appelait "paquets" – à travers un réseau numérique sur le principe du stockage et retransmission. En 1966, après l'annonce de son travail précurseur sur la commutation par paquets, il fut nommé chef de la division informatique du NPL. Les motifs qui avaient conduit Davies à l'idée d'un réseau de commutation par paquets n'avaient rien à voir avec les préoccupations militaires qui avaient poussé Baran. Davies voulait simplement créer un nouveau réseau public de communication. Il a prévu la nécessité de maîtriser la diversité des matériels et des logiciels, autrement dit les différences séparant les langages informatiques ou les systèmes d'exploitation des machines. Contrairement à la réaction très fraîche d'AT&T à l'égard de Baran, les télécommunications britanniques ont épousé les idées de Davies. Cela l'a encouragé à chercher un financement pour bâtir un réseau expérimental au NPL (voir [HL–96], pp.78–81).

La théorie de la commutation par paquets a été également étudiée par Leonard Kleinrock en 1959 lorsqu'il était étudiant de troisième cycle au MIT ([KLE–61], [KLE–64]). Il effectua un travail théorique important qui décrivait une série de modèles analytiques pour les réseaux de communication.

1.4.6 Le premier réseau : ARPANET en 1971

Le vendredi 4 octobre 1957, l'Union Soviétique lance le premier satellite artificiel, appelé Spoutnik. Les Américains considèrent cela comme une menace. Le président Einsenhower demande le 7 janvier 1958 au Congrès les fonds nécessaires pour créer l'ARPA (*Advanced Research Projects Agency*). Directement lié au président at au secrétaire à la défense, cet organisme de recherche avait pour but de garantir, par la promptitude de sa réaction, que les américains ne

soient désormais plus jamais en retard dans aucun domaine technologique ([HL–96], pp. 19–31).

En 1961, le directeur de l'ARPA cherchait quelqu'un pour gérer le contrat d'un nouvel ordinateur commandé par l'ARPA, le Q-32, ainsi que quelqu'un qui pût diriger un nouveau programme, demandé par le département de la Défense et axé sur les sciences du comportement. À l'automne 1962, il trouve enfin un candidat susceptible d'occuper les deux postes, un éminent psychologue nommé Joseph Carl Robnett Licklider. D'après son contrat, sa tâche principale était de trouver pour l'ordinateur des utilisations qui en fassent autre chose qu'un outil destiné aux calculs numériques des scientifiques. En un rien de temps, il prit contact avec les meilleurs informaticiens du moment, à Stanford, au MIT, à Berkeley, à UCLA (*University of California, Los Angeles*), ainsi qu'avec une poignée de sociétés. Six mois après son arrivée à l'ARPA, Licklider envoya une longue note aux membres de cet entourage à propos de la dispersion excessive des thèmes de recherche ([HL–96], pp.32–49). Comment parvenir à y remédier ? Il discuta de l'hypothèse d'un réseau (*network*) d'ordinateurs :

> "Considérez la situation où plusieurs centres sont réunis dans un même filet [*netted*], chaque centre étant tout à fait singulier, avec son propre langage et sa propre façon de façon de faire les choses. N'est-il pas désirable ou même nécessaire que tous les centres s'accordent sur un quelconque langage ou, du moins, sur quelques conventions pour poser des questions telles que : "Quel langage parlez-vous ?" À ce point, le problème est avant tout celui dont débattent les auteurs de science-fiction : comment amorcer des communications entre des êtres doués de raison, mais privés de toute forme de correspondance ?"

L'une des personnes récrutées par Licklider, Robert Taylor, devint le directeur du service de celui-ci, le LIPTO, en 1966. Il décida de mettre en pratique les idées de Licklider et demanda un financement pour faire l'expérience d'un réseau d'ordinateurs. Il suggéra que l'ARPA finance un petit réseau à titre d'essai : on commencerait, par exemple, avec quatre nœuds, et on poursuivrait jusqu'à une douzaine environ ([HL–96], pp.49–53). Il obtînt un million de dollars pour mettre en place son système.

L'architecte d'un tel réseau devrait être également un expert en télécommunication. Taylor recruta Larry Roberts en décembre 1966 ([HL–96], pp.55–63). Celui-ci commença par écrire une note dans laquelle il appelait les ordinateurs intermédiaires qui contrôleraient le réseau des "serveurs de messages", des **IMP** pour *Interface Message Processor*. Ils devaient remplir les fonctions suivantes : interconnecter le réseau, envoyer et recevoir des données, effectuer des tests d'erreur, retransmettre en cas d'erreur, acheminer les données et vérifier que les messages arrivent aux destinations voulues ([HL–96], pp.81–91). Un protocole serait établi pour définir avec exactitude comment les IMP devraient communiquer avec les ordinateurs hôtes (il ne le sera jamais).

Fin 1967, Roberts présenta son premier exposé sur ce qu'il appela l'"ARPA net", le réseau de l'ARPA, à un colloque d'informatique de l'ACM à Gatlinburg, dans le Tennessee.

Roberts estimait que le réseau devait démarrer avec quatre sites : UCLA, le SRI (*Stanford Research Institute*), l'université d'Utah et l'UCSB (*University of California, Santa Barbara*). Dans un second temps, il se développerait jusqu'à en réunir dix-neuf environ. Il décida de passer un appel d'offres, qu'il termina fin juillet 1968. Les premières réactions, négatives, à l'appel d'offres vinrent des deux plus grands constructeurs d'ordinateurs, IBM et Control Data Corporation (CDC). Les deux sociétés refusèrent de soumissionner car elles estimaient que le réseau ne pourrait jamais être construit parce qu'il n'existait pas d'ordinateur assez petit pour rendre l'affaire rentable. Plus d'une douzaine d'offres furent soumises. Juste avant noël 1968, l'ARPA annonça que le contrat pour la construction des serveurs de messages était attribué à Bolt, Beranek & Newman (BBN), une petite société de conseils de Cambridge, Massachussetts ([HL–96],

pp.92–97).

Les cinq années suivantes furent consacrées à des tests et à des mises au point. En 1971, ARPANET entra en service régulier. Les machines utilisaient ARPANET en se connectant à un IMP. La façon de connecter un IMP au réseau se faisait à l'aide du **protocole "1822"**, dont le nom provient du nombre de pages techniques décrivant le système. Nous verrons que la façon dont un hôte devait se relier à l'IMP ne fut jamais décrit dans un protocole de la part des concepteurs de l'ARPANET.

Notons qu'en 1980 l'agence change de nom, passant de ARPA à DARPA (*Defense Advanced Research Projects Agency*).

Références

[TAN–81], à travers ses éditions successives, est devenu le classique pour une vue d'ensemble sur les réseaux. Il contient beaucoup d'informations mais pas d'exemple étudié en détail. [K-R–01] peut devenir un concurrent sérieux et au minimum un très bon complément. [PUJ–04] est le classique français.

La norme ISO 7498, *Open System Interconnection Model standard* est décrite dans quatre documents, disponibles en anglais et en français : [ISO 7498-1], [ISO 7498-2], [ISO 7498-3] et [ISO 7498-4].

L'architecture TCP/IP a été décrite, après coup, dans [RFC 1122].

Première partie

Couches physique et de liaison

Nous avons décidé, après mûre réflexion, d'inclure cette première partie dans cet ouvrage (consacré à l'implémentation du sous-système réseau de Linux) à titre de complétude sur l'étude des réseaux. Elle ne joue aucun rôle dans le sous-système réseau des systèmes d'exploitation. Elle peut donc être passée sans problème pour la lecture de la suite.

Chapitre 2

La couche physique

La *couche physique* est chargée de la transmission des bits à l'état brut sur un canal de communication. Les problèmes de conception concernent les interfaces mécaniques (le nombre de broches d'un connecteur et leur rôle) et électriques (le nombre de volts à fournir pour représenter un 1 et un 0), la synchronisation ainsi que le support physique de transmission (*medium* en anglais). La prise en compte de cette couche concerne donc le matériel, les concepteurs des cartes réseau, des micro-ordinateurs (et des routeurs) et non les logiciels que sont les sous-systèmes réseau des systèmes d'exploitation. Par conséquent Linux n'est pas partie prenante de la couche physique.

Nous allons cependant mettre en avant un minimum de connaissances concernant la couche physique qui nous serons utiles, sans être à proprement nécessaires, pour la suite.

2.1 Les supports physiques

Un bit peut être représenté par une donnée mécanique, électrique (par une certaine différence de potentiel), hydraulique, pneumatique (c'est-à-dire par une certaine pression), optique ou autre. Actuellement, et sauf à titre tout à fait expérimental, les bits sont toujours représentés au niveau de l'ordinateur électroniquement en jouant sur une certaine différence de potentiel. Ce n'est pas le cas dans les canaux de communication où l'on utilise une fréquence (cas du réseau téléphonique ou des lignes spécialisées), des ondes radio ou des rayons lumineux (dans le cas des fibres optiques). Cependant les signaux sont toujours transformés à l'arrivée et, au niveau de l'ordinateur, on n'a à tenir compte que de différences de potentiel.

Nous renvoyons, par exemple, au chapitre deux de la dernière édition en cours de [TAN–81] pour une vue d'ensemble des medias physiques utilisés pour la couche physique. Pour ce qui nous concerne, nous allons nous contenter de les citer :

- les supports avec guide physique (*transmission filaire*) :
  - avec une âme en cuivre comprennent les paires torsadées et les câbles coaxiaux ;
  - remplacés pour les distances moyennes par des fibres optiques ;
- les supports sans guide physique (*transmission sans fil*) utilisent les ondes radio :
  - *via* des relais hertziens ;
  - ou des satellites de communication.

2.2 Interface support physique/ordinateur

Le modèle que l'on peut prendre pour l'interface entre le support transportant les données réseau et l'ordinateur est un flux de bits, comme dans le cas d'une liaison série. La liaison série fut d'ailleurs longtemps utilisée, par exemple dans le cas d'un accès au réseau *via* un modem (avant l'arrivée du câble et de ADSL).

2.2.1 Rappels sur les liaisons série

Nous nous permettons de renvoyer au chapitre 24 de notre livre [CEG–03] pour une étude des liaisons série. Rappelons-en ce qui est essentiel pour la suite.

Dans une **liaison série**, les données sont transmises un bit à la fois (et non pas un ou plusieurs octets à la fois). On distingue deux types de communications séries : les *communications séries asynchrone* et les *communications séries synchrones*.

2.2.1.1 Liaisons série asynchrone

Dans une **communication série asynchrone**, les octets sont transmis de façon irrégulière ; il faut donc un **marqueur** pour indiquer le début de l'envoi d'un certain nombre de bits, convenu à l'avance, et un marqueur pour en indiquer la fin.

Une communication série asynchrone se caractérise par la nature et le nombre de ses bits de début et d'arrêt et par son taux de transmission :

Bit de début et bit d'arrêt. La technique courante pour les communications série asynchrones consiste à maintenir le signal à un niveau haut (une **marque**, en anglais *mark*) jusqu'à ce qu'une donnée soit transmise. Chaque caractère transmis doit commencer par un bit de niveau 0 (un **espace**, en anglais *space*), appelé le **bit de début** (en anglais *start bit*). Il