Utilisation des iptables

Patrick Cegielski
cegielski@u-pec.fr

Février 2017

Pour Irène et Marie

Legal Notice

Université Paris XII - IUT Sénart-Fontainebleau
Route forestière Hurtault
F-77300 Fontainebleau
cegielski@u-pec.fr

Table des matières

Table des figures

Première partie

Introduction aux réseaux

Chapitre 1

Modèle et architecture des réseaux informatique

1.1 Apparition des réseaux informatique

Au milieu des années 1960 vint l'idée de relier plusieurs unités centrales entre elles et non plus seulement une unité centrale et des *terminaux*.

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.

1.1.1 Naissance des ordinateurs (1949)

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* (*computer* en anglais), 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].

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.

Exercice 1.- *Procurez-vous un livre d'histoire de l'informatique et lisez-le.*

1.1.2 Les réseaux de communication

De nos jours, « réseau » sans adjectif désigne toujours un réseau informatique. Cependant 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* permettent le transport de matériels. On distingue les communications terrestres (par route), fluviaux, maritimes, ferroviaires et aériennes (par avion). Les premiers réseaux routiers cohérents commencent avec les Romains, revus aux dix-neuvième siècle avec le réseau de routes nationales (départementales et vicinales) et celui des autoroutes au vingtième siècle.

Les *réseaux de télécommunication*, c'est-à-dire sans déplacement de matière, commencent avec les feux dans l'Antiquité, le tam-tam en Afrique ou les signaux de fumée des indiens d'Amérique. Il prennent vraiment leur essor avec le télégraphe optique de Chappe sous la Révolution puis avec le télégraphe électrique qui le détrônera dans la seconde moitié du dix-neuvième siècle, le début des premiers câbles transatlantiques, le téléphone, la T.S.F. (*Télégraphie Sans Fil* puis *Téléphonie Sans Fil*), la télévision et, pour finir, les réseaux informatiques qui nous intéressent.

Exercice 2.- *Renseignez-vous sur la construction des routes et surtout sur le développement d'un réseau cohérent de routes, en particulier grâce à une série de publications des* Presses de l'École des Ponts-et-Chaussées.

Exercice 3.- *Renseignez-vous de même sur le développement d'un réseau cohérent de communications maritimes, fluviales, ferroviaires ou aériennes.*

Exercice 4.- *Renseignez-vous sur le principe du télégraphe optique et sur le développement du réseau de celui-ci.* On pourra partir de l'article « Télégraphe Chappe » de *Wikipédia.*

Exercice 5.- *Renseignez-vous sur le principe du télégraphe électrique et sur le développement du réseau de celui-ci.* On pourra partir de l'article « Télégraphe » de *Wikipédia.*

Exercice 6.- *Renseignez-vous sur le principe du téléphone et sur le développement du réseau de celui-ci.* On pourra partir de l'article « Téléphone » de *Wikipédia.*

Exercice 7.- *Renseignez-vous sur le principe de la T.S.F. et sur le développement du réseau de celle-ci.* On pourra partir de l'article « Télégraphie sans fil » de *Wikipédia.*

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'états et non des machines de Turing.

1.1.3 Les terminaux distants (1952)

L'alliance des télécommunications et de l'informatique donne naissance à ce qui fut appelé un temps *télématique*, abréviation de *TÉLÉcommunication* et d'*inforMATIQUE*.

En effet le coût très onéreux d'une unité centrale d'un ordinateur a conduit à partager celle-ci entre plusieurs sites, d'une université par exemple. pour des raisons de bonne économie. Des raisons intrinsèquement liées à l'application envisagée ont permis de développer cet aspect : le projet SAGE (*Semi-Automatic Ground Environment*, environnement au sol semi-automatique) de surveillance radar des États-Unis, opérationnel de 1952 à 1984, ou son pendant civil SABRE (déformation plus familière de SABER [*Semi-Automatic Business Environment Research*], soit projet de « Recherche pour un environnement opérationnel semi-automatique ») de réservation des billets d'avion d'*American Airlines*, déployé en 1960 et toujours utilisé (en particulier par la SNCF en France).

Il s'agit de l'utilisation de *terminaux* distants de l'unité centrale, parfois situés à plusieurs milliers de kilomètres.

Le premier problème technique à résoudre pour cela est celui de la prise en charge des utilisateurs multiples (*time-sharing* en anglais), c'est-à-dire de plusieurs utilisateurs reliés à une même unité centrale, ayant chacun l'impression d'être le seul à l'utiliser. C'est devenu de nos jours 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. CTSS est utilisé dans la série des ordinateurs IBM 7090/94. On a alors la notion d'**accès à distance** (*remote access*) à un **ordinateur hôte** (*host computer* en anglais).

Ceci conduit, en 1964, un groupe de chercheurs du MIT, des *Bell Laboratories* et de *General Electric* à s'associer pour initier le développement d'un ambitieux système d'exploitation multi-utilisateurs, qu'ils baptisent projet **MULTICS** (*Multiplexed Information and Computing System*). Ce projet n'aboutit pas et est abandonné en 1968 mais il est à l'origine du système UNIX, qui date de 1971.

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

Exercice 8.- *Renseignez-vous sur le système SAGE.* On pourra partir de l'article « Semi-Automatic Ground Environment » de *Wikipédia*, en français pour commencer mais surtout en anglais.

Voir ou revoir le film *Docteur Folamour* de Stanley KUBRICK (1964).

Exercice 8.- *Renseignez-vous sur le système SABRE.* On pourra partir de l'article « Sabre (informatique) » de *Wikipédia*, en français et éventuellement en anglais.

1.1.4 Première mise en réseau (1965)

On parle de **réseau informatique** lorsqu'on met en relation au moins deux ordinateurs, non une unité centrale et un terminal, et pas non plus deux exemplaires d'un ême type d'ordinateurs mais des ordinateurs vraiment différents.

La première mise en réseau, en ce sens, est effectuée en 1965.

Le psychologue Tom (Thomas) MARRILL lance cette année-là une petite société de systèmes en temps partagé, au sens où nous l'avons vu dans la section précédente. Son principal investisseur lui

fait défaut à la dernière minute ; il doit donc chercher un contrat de recherche et développement. Il propose à l'ARPA, dont nous allons reparler dans un instant, 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 MARRILL est si petite que l'ARPA lui recommande de procéder à son expérience sous l'égide du *Lincoln Laboratory*. L'idée plait aux responsables du *Lincoln* et ils chargent Larry ROBERTS de surveiller le projet.

La liaison entre les deux ordinateurs est réalisée grâce à un service spécial de la *Western Union* : quatre fils en duplex intégral. MARRILL branche sur cette liaison un type de **modem** (*modulateur/démodulateur*) rudimentaire, opérant à 2 000 bits par seconde, qu'il appelle un composeur automatique (*automatic dialer*).

MARRILL conçoit également une procédure pour grouper les caractères en messages, les envoyer et vérifier qu'ils sont arrivés. Si aucun accusé de réception ne suit, le message est transmis à nouveau. Il appelle « protocole » de message l'ensemble des procédures destinées à faire circuler l'information dans les deux sens.

En dépit de leurs efforts, lorsque MARRILL et ROBERTS connectent effectivement les deux machines, le résultat est mitigé : le temps de réponse est médiocre ([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. Ils écrivent un rapport [M-R-66] décrivant l'expérience.

> LES RÉSEAUX ET LE PROBLÈME DE L'INCOMPATIBILITÉ DES ORDINATEURS
>
> Les machines incompatible représentent un problème déjà ancien en informatique. Très souvent, à cause de l'incompatibilité des ordinateurs, des programmes développés pour une installation ne sont pas disponibles pour d'autres installations. Le même programme peut ainsi devoir être réécrit des douzaines de fois. Supposons, par exemple, qu'un programme effectuant l'analyse syntaxique de phrases en Anglais naturel ait été développé pour un certain ordinateur, Y. Un tel programme devrait intéresser les chercheurs en entrées en langue naturelle pour les ordinateurs, en traduction automatique [...]. Malheureusement, le programme ne sera disponible qu'à ceux qui ont un accès direct à Y (ou à un ordinateur compatible avec Y) ; les utilisateurs des autres installations devront coder à nouveau le programme pour leurs propres ordinateurs, avec un coût par ordinateur comparable au coût du développement originel. Vues à l'échelle d'une nation, de telles inefficacités peuvent devenir très onéreuses.
>
> Les remèdes proposés pour l'incompatibilité des ordinateurs sont à ce jour les suivants.
>
> 1. Utiliser des ordinateurs identiques. Ainsi, par exemple, l'utilisation de deux IBM 7094 modifés identiques pour le projet MAC et le Centre de Calcul du MIT a rendu possible le développement du *Compatible Time-Sharing System* (CTSS), permettant aux programmes écrits pour l'un des système de tourner sur l'autre.
>
> 2. Écrire des programmes dans un langage de haut niveau pour lequel des compilateurs existent sur les différentes machines. Ainsi, un programme source donné écrit en FORTRAN peut être compilé et tourner sur un grand nombre d'ordinateurs différents.
>
> Malheureusement, ces remèdes n'ont pas très bien marché dans le passé et ne marcheront probabement pas mieux dans les environnements à temps partagé du futur. En ce qui concerne le remède (1), il n'existe pas de signe montrant que la

prolifération du matériel cesse. Il n'est pas difficile de trouver des exemples de situations dans lesquels, même dans une même organisation, de nouveaux ordinateurs incompatibles sont ajoutés à ceux déjà en place, exigeant la duplication des efforts de programmation. En ce qui concerne le remède (2), on voit de nouveaux langages de programmation développés chaque jour. [...]

Ainsi nous ne voyons aucune raison de croire que les vieux remèdes marcherons mieux dans le futur que dans le passé. Il est probable, cependant, que la technique des réseaux informatique (*computer networking*) contribue à la solution du problème. [...] Pour revenir à l'exemple ci-dessus, si un utilisateur de la machine X veut utiliser, comme partie d'un programme plus conséquent, la routine d'analyse syntaxique tournant sur l'ordinateur Y, ce programme plus conséquent devra communiquer la phrase à analyser à Y, demander au programme d'analyse syntaxique de s'exécuter sur Y, de récupérer les resultats et continuer sur X.

CONSIDÉRATIONS LOGICIELLES

L'approche élémentaire

Nous commencerons par une approche élementaire du problème de constitution des réseaux informatiques. Cette approche ne représente certainement pas la meilleure alternative, mais elle a l'avantage comme point of départ d'être de loin la plus simple, puisqu'elle permet à un réseau de systèmes à temps partagé existant à se transformer en réseau informatique sans changement appréciable de matériel ou de logiciel le pilotant.

En utilisant cette approche, la seule contrainte que doive supporter le système pour pouvoir se relier au réseau est la suivante : le pilote de temps partagé doit permettre aux programmes des utilisateurs de communiquer avec deux terminaux. [...]

En parlant métaphoriquement, nous pouvons considérer le lien d'un ordinateur à l'autre, dans un tel réseau, comme étant le résultat de l'extirpation d'un terminal utilisateur de son câble le reliant à l'ordinateur X, l'extirpation d'un terminal utilisateur de son câble le reliant à l'ordinateur Y, et de relier ces deux câbles ensembles.

Un tel réseau fonctionne comme suit. L'utilisateur appelle son ordinateur hôte, CH, à partir d'une console. Il se logue normalement en transmettant des caractères depuis sa console au moniteur MH. Il installe son programme utilisateur PH et démarre ce programme. PH, grâce au second canal disponible, se logue à l'ordinateur distant CR, en transmettant la suite correcte de symboles au moniteur distant MR. (Remarquez que c'est le programme de l'utilisateur PH, non le moniteur MH, qui a la responsibilité de faire ceci.) Le moniteur distant MR prend en charge les actions et communique à PH que les actions ont été exécutées. PH prend en considération ces messages. PH communique alors avec MR pour installer le programme désiré PR sur l'ordinateur distant ; il exécute alors PR et transmet et accepte des données de celui-ci, jusqu'à ce que soit fait. PH se délogue alors de MR. PH continue jusqu'à ce que la tâche soit terminée. L'utilisateur se délogue de MH.

Remarquez que ni MH, ni MR n'ont besoin de se comporter de façon inhabituelle. Les moniteurs font ce qu'ils ont toujours fait. La seule exigence, comme nous l'avons dit ci-dessus, est que le programme de l'utilisateur PH puisse communiquer avec deux terminaux, son propre terminal utilisateur et l'ordinateur distant. La plupart des moniteurs actuels ont cette possibilité.

[...]

EXPÉRIMENTATION DU RÉSEAU

Un travail est en cours sur l'implémentation d'un réseau expérimental impliquant le système de temps partagé APEX tournant sur l'ordinateur TX-2 du laboratoire Lincoln du MIT de Lexington, Massachusetts, et le système à temps partagé tournant sur l'ensemble d'ordinateurs Q-32/PDP-1 à la *System Development Corporation* de Santa Monica, Californie. Initialement, un système d'appel 4KC à quatre fils sera utilisé avec des modems asynchrones de 1200 bits par seconde. Le protocole de messages donné dans l'appendice sera utilisé. [...]

APPENDICE

PROTOCOLE DE MESSAGES POUR LE LIEN TX-2/Q-32

Cet appendice décrit le protocole de messages à utiliser pour le lien entre le Q-32 de la *System Development Corporation* de Santa Monica, Californie, et le TX-2 du laboratoire Lincoln à Lexington, Massachusetts.

Chaque caractère est formé de huit bits de données, envoyés avec le bit de plus faible poids en premier, précédé d'un bit de début égal à zéro et suivi d'un bit de fin égal à un. Lorsqu'il ne transmet pas de caractère, le lien transmet continuement un un.

Tableau 1. Caractères spéciaux pour le protocole de message

Octal	ASCII	Meaning
EN-TÊTE		
201	SOH	caractères pour le moniteur
202	STX	caractères pour l'utilisateur
221	DCI	données pour le moniteur
232	SS	données pour l'utilisateur
FIN DE MESSAGE		
203	ETX	fin de message
ACCUSÉ DE RÉCEPTION		
225	NACK	erreur dans le messager, répéter
234	FS	message OK, mais attendre
206	ACK	message OK, envoyer message suivant
REQUÊTE		
230	CNCL	renvoyer le dernier accusé de réception
SYNCHRONISATION		
226	SYNC	ignorer
FONCTIONS SPÉCIALES		
220	DLE	aide/pause
233	ESC	panique.

Toutes les information transmises sont envoyées sous la forme de messages constitués d'un caractère d'en-tête, d'un corps, d'un caractère de fin de message et d'une somme de contrôle. Tous les messages reçoivent un accusé de récption.

Il y a quatre types de messages. Chacun a un unique caractère d'en-tête qui détermine à la fois la destination du message (utilisateur ou moniteur) et le mode du message (chaîne de caractères ou donnée binaire). Les caractères spécifiques utilisés sont listés dans le Tableau 1.

> Le corps du message a une longueur maximum de 119 caractères si le message est une chaîne de caractères et de 118 caractères si le message est une donnée binaire. Si le message est constitué de données binaires, le premier caractère du corps est un caractère de dénombrement égal au nombre total de caractères dans le corps, y compris le caractère de dénombrement. Les valeurs valides vont de deux à 118.
>
> Le corps du message est suivi par un caractère de fin de message. Celui-ci est suivi immédiatement par une somme de contrôle, la somme sur 8 bits du caractère d'en-tête et de tous les caractères du corps.
>
> Tout message reçoit un accusé de réception constitué d'un caractère parmi trois possibles. Le premier indique une erreur, demandant la retransmission. Le second indique que le message a été reçu correctement, mais demande que l'expéditeur attende avant d'envoyer le message suivant, car le tampon du destinataire est plein. Le troisième type d'accusé de réception indique que le dernier message a été reçu correctement et/ou que le destinataire est prêt pour le message suivant.
>
> Il y a quatre autres caractères spéciaux. Le premier, requête, demande au destinataire de renvoyer le dernier caractère d'accusé de réception qu'il a reçu, ou indique une erreur s'il est en train de recevoir un message ou des déchets depuis le dernier message correct.
>
> Le second, sync, sera utilisé par un autre lien (synchrone) attaché au TX-2. En ce qui concerne le lien SDC/TX-2, les caractères de synchronisation sont ignorés. Les deux autres, aide et panique, sont tous les deux traités comme une rupture pour le SDC ou aident aux requêtes pour Lincoln. Plus tard le caractère de panique sera utilisé pour une interruption d'ordre supérieur pour Lincoln.
>
> Comme décrit ici, le système est capable de transmettre et de recevoir des messages simultanément. Pour accélérer la transmission du texte, les accusés de réception et les requêtes peuvent être placées au milieu des chaînes de caractères (mais pas des données binaires), de sorte que le destinataire devra toujours se préoccuper des caractères spéciaux en mode caractère. Ces caractères spéciaux ne sont pas inclus dans la somme de contrôle. [...]
>
> Pour éviter les « décrochages », l'expéditeur est responsable de l'envoi des messages et des accusés de réception. Si l'expéditeur ne reçoit pas l'accusé de réception attendu dans la seconde, une requête est envoyée pour voir si le message ou l'accusé de réception est perdu. De même, si une condition « prêt » (ACK) n'est pas reçue dans les 30 secondes après un « attendre » (FS), une requête est envoyée pour voir si ce « prêt » a été perdu.
>
> Tous les caractères spéciaux ont leur bit de plus haut poids égal à un. Ceci laisse les 128 caractères dont le bit de plus haut poids est nul disponibles pour transmettre en utilisant le code ASCII à 7 bits en mode caractère. [...]

Profitons-en pour dire ce qu'est l'ARPA, puisque cette agence est fortement liée au développement des réseaux. 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*). Dépendant directement du président et du 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). Cette agence employait une centaine de scientifiques de haut niveau avec un budget suffisant pour passer des contrats avec d'autres institutions.

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

1.1.5 Le premier réseau (ARPAnet, 1971)

En 1961, le directeur de l'ARPA cherche quelqu'un pour gérer le contrat d'un nouvel ordinateur commandé par l'ARPA, le Q-32, ainsi que quelqu'un qui puisse diriger un nouveau programme, demandé par le département de la Défense et axé sur les sciences du comportement. À l'automne 1962, il trouve un candidat susceptible d'occuper les deux postes, un éminent psychologue nommé Joseph Carl Robnett LICKLIDER. D'après son contrat, sa tâche principale est de trouver pour l'ordinateur des utilisations qui en fasse autre chose qu'un outil destiné aux calculs numériques des scientifiques. En un rien de temps, il prend contact avec les meilleurs informaticiens du moment, à Stanford, au MIT, à Berkeley et à l'UCLA (*University of California, Los Angeles*), ainsi qu'avec quelques sociétés. Six mois après son arrivée à l'ARPA, LICKLIDER envoie 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 pense alors à 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 faire les choses. N'est-il pas désirable, et même nécessaire, que tous ces 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 recrutées par LICKLIDER, Robert TAYLOR, devient le directeur du service chargé de ce projet, appelé LIPTO, en 1966. Il décide de mettre en pratique les idées de LICKLIDER et demande un financement pour faire l'expérience d'un réseau d'ordinateurs. Il suggère 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 obtient un million de dollars pour mettre en place son système.

L'architecte d'un tel réseau doit être un expert en télécommunications. TAYLOR recrute Larry ROBERTS en décembre 1966 ([HL–96], pp.55–63). Celui-ci commence par écrire une note dans laquelle il appelle les ordinateurs intermédiaires qui contrôlent le réseau des « serveurs de messages », des **IMP** (pour *Interface Message Processor*). Ceux-ci doivent 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 doit être établi pour définir avec exactitude comment les IMP doivent communiquer avec les ordinateurs hôtes (il ne le sera jamais).

Fin 1967, ROBERTS présente son premier exposé sur ce qu'il appelle l'« ARPA net », le réseau de l'ARPA, à un colloque d'informatique de l'ACM (*Association for Computing Machinery*)qui se tient à Gatlinburg, dans le Tennessee [ROB-67].

ROBERTS estime que le réseau doit démarrer avec quatre sites : l'UCLA, le SRI (*Stanford Research Institute*), l'université de l'Utah et l'UCSB (*University of California, Santa Barbara*). Dans un second temps, il doit se développer jusqu'à en réunir dix-neuf environ. Il décide de passer un appel d'offres, dont il termine la rédaction fin juillet 1968. Les premières réactions, négatives, à l'appel d'offres viennent des deux plus grands constructeurs d'ordinateurs, IBM et CDC (*Control Data Corporation*). Les deux sociétés refusent de soumissionner car elles estiment que le réseau ne pourra jamais être construit, parce qu'il n'existe pas d'ordinateur assez petit pour rendre l'affaire rentable. Plus d'une douzaine d'offres sont cependant soumises. L'ARPA choisit l'UCLA pour le *Network Measurement Center* en octobre. Juste avant noël 1968, l'ARPA annonce que le contrat pour la construction des serveurs de messages (les IMP) est attribué à

BBN (*Bolt, Beranek & Newman*), une petite société de conseils de Cambridge, Massachussetts, dirigée par Frank HEART ([HL–96], pp.92–97). BBN a un million de dollars et moins d'un an pour transformer la théorie en quelque chose qui fonctionne. Elle choisit un mini-ordinateur Honeywell comme base de l'IMP.

Les cinq années suivantes sont consacrées à des tests et à des mises au point. Le 2 septembre 1969, BBN termine l'installation du premier IMP (IMP1) à l'UCLA, attaché au Sigma-7 de SDS de l'école : le choix du premier site est dû au développement de la théorie de la commutation par paquets de KLEINROCK. Le premier octobre, le second IMP est installé au SRI, connecté à leur SDS 940 : le choix est dû au projet de Doug ENGELBART sur l'« augmentation de l'intellect humain » (qui comprend NLS, un système hypertexte). Après quelques essais, la première connexion est réalisée de l'UCLA au SRI *via* une ligne AT&T de 50 kbps. Les premiers paquets sont envoyés par Charley KLINE de l'UCLA pour essayer de se loguer au SRI : le système s'effondre lorsque la lettre G de LOGIN est entrée. IMP3 est installé à l'UCSB en novembre et IMP4 à l'université de l'Utah en décembre. IMP5 est installé au quartier général de BBN à Boston en mars 1970.

En 1971, ARPANET entre en service régulier. 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.

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 d'installations téléphoniques. Au début, les téléphones étaient vendus par paire et il appartenait au client de les relier à l'aide d'un couple de fils électriques (puis un seul, puisqu'on s'aperçut très vite que la voie du retour peut se faire par la terre). Si une personne souhaitait pouvoir converser avec n 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.

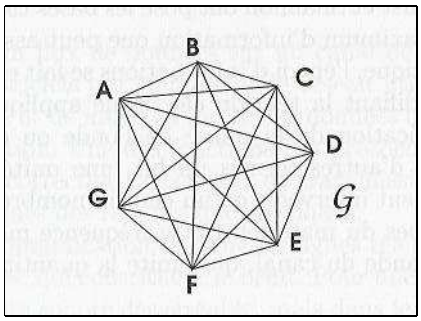


Figure 1.1 – Graphe complet

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.

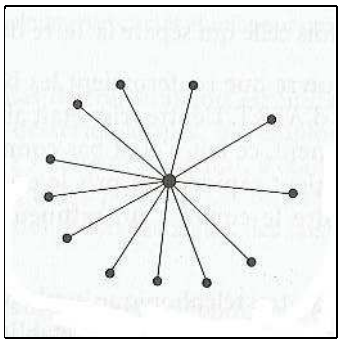


Figure 1.2 – Réseau avec commutateur central

Bell réagit à cette situation en fondant la société *Bell Telephone Company*, qui crée la notion de **central téléphonique** et met le premier central en place à New Heaven (Connecticut) en 1878. La société amène un fil électrique à chaque domicile ou bureau d'*abonné* : pour effectuer

un appel, le client tourne une manivelle, ce qui active une sonnerie au niveau du central téléphonique, attirant l'attention d'une opératrice ; celle-ci se charge 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.

Très rapidement d'autres centraux sont créés un peu partout. Comme des clients souhaitaient appeler des correspondants dans d'autres villes que la leur, il a fallu interconnecter ces centraux, donnant lieu à l'*interurbain*. C'est, 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 contacte 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 entre l'appelant et l'appelé. Un tel chemin est appelé un **circuit** dans le jargon de la téléphonie. On parlera plus tard de **commutation de circuits** (*circuit switching* en anglais) pour ce type de commutation lorsque d'autres types de commutation seront apparus. 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 est établie manuellement, comme nous venons de le voir. On parlera plus tard de **commutation manuelle**. Peu après l'invention du téléphone, un équipement **automatique** de **commutation** électro-mécanique est inventé par Almon Strowger, qui sera utilisé pendant près de cent ans : pour la petite histoire, il s'agit d'un organisateur de pompes funèbres qui en avait assez que la femme d'un de ses collègues, opératrice téléphonique de son état, passe toutes les communications désespérées des membres de la famille d'un défunt à son collègue et non à lui. Au début des années 1970, il est remplacé par un équipement électronique, mais le principe est le même pour ce qui nous intéresse.

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 ne l'est pas pour un réseau informatique.

— Il peut arriver que toutes les liaisons d'un central à l'autre soient occupées. Un « disque » demande alors de renouveler l'appel ultérieurement.

— Le débit du circuit 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 qui servaient à transmettre les télégrammes.

L'énorme avantage, par rapport à la commutation de circuits, 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 empêchent 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 de grande capacité 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, chacun d'eux possédant une taille maximale. Ceci évite d'avoir à utiliser des disques tampon d'une grande capacité, d'une part, et la congestion durant plusieurs minutes due à un gros message, d'autre part.

Les paquets empruntent éventuellement des chemins différents, s'il en existe plusieurs, ce qui peut entraîner des arrivées désordonnées. On doit donc numéroter 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.

1.2.2.4 Cas des réseaux informatiques

Dans le cas des réseaux informatiques, on utilise le plus souvent la commutation par paquets (c'est le cas pour Internet) et plus rarement la commutation de circuits (c'est le cas d'ATM [*Asynchronous Transfer Mode*, soit « mode de transfert asynchrone »]) 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**. Cette façon de faire est prônée par Paul Baran avant de devenir réalité lors de la réalisation des réseaux informatiques.

Paul Baran s'intéresse 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 sont extrèmement vulnérables et hors d'état de supporter une attaque nucléaire. Pourtant le pouvoir qu'a le président des États-Unis de commander ou d'annuler l'envoi de missiles nucléaires repose sur ces systèmes de communications vulnérables. Baran est l'un des premiers à établir, au moins de façon théorique, que le problème peut être résolu. Il en est arrivé à l'idée qu'un réseau de transmission de données peut 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 est possible d'installer un réseau avec de nombreuses connexions redondantes. Mais il y a une restriction technique : tous les signaux sur le réseau téléphonique sont analogiques. Le plan d'acheminement sur ce réseau interdit 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].

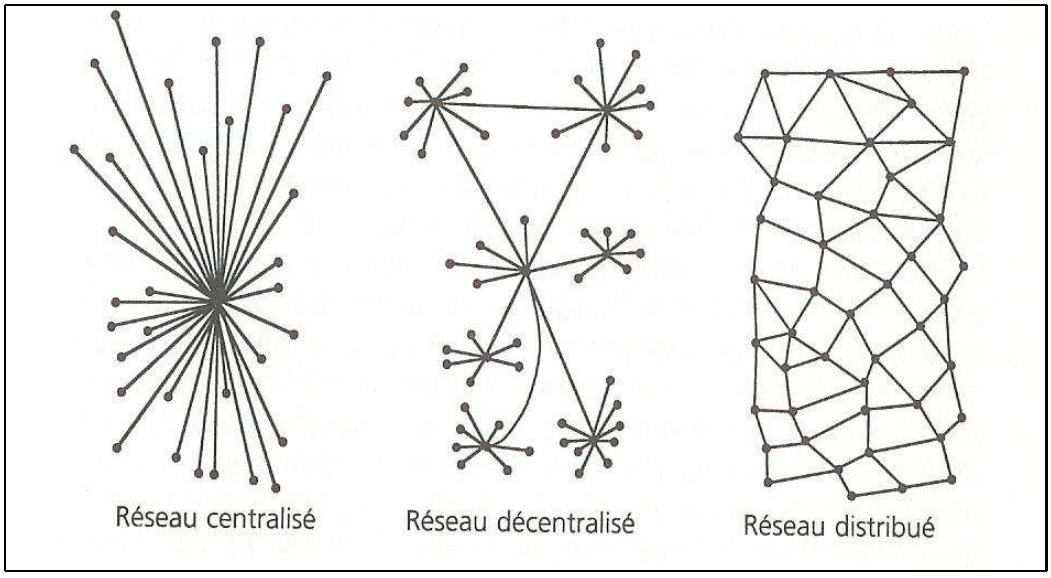


Figure 1.3 – Réseau distribué [BAR–64]

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

1.2.4 Historique de la commutation par paquets

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

Nous venons de voir comment Paul Baran en est venu à l'idée de l'utilisation des réseaux maillés. La seconde contribution de Baran aux réseaux est plus révolutionnaire encore : disloquer les messages pour obtenir la commutation par paquets, qu'il appelle « blocs-messages ». Dans le modèle de Baran, chaque nœud de communication contient une *table de routage* qui se comporte comme une sorte de coordinateur ou de régulateur. Il espère persuader AT&T des avantages de son projet mais ce n'est pas le cas. En 1965 cependant, cinq ans après s'être lancé dans le projet, Baran obtient l'appui complet de la RAND, qui veut construire un réseau de commutation distribué. Malheureusement AT&T, qui dispose du monopole des télécommunications, s'y oppose. Baran se tourne alors vers d'autres projets ([HL–96], pp.72–78).

À Londres, à l'automne 1965, juste après que Baran laisse tomber son projet, Donald Watts Davies, quarante et un ans, physicien au *British National Physical Laboratory* (NPL), écrit une première note exposant ses idées à propos d'un nouveau type de réseau, fort proche de celui de Baran. Le printemps suivant, il donne à Londres une conférence publique dans laquelle il décrit l'envoi de petits blocs de données – qu'il appelle « 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 est nommé chef de la division informatique du NPL. Les motifs qui ont conduit Davies à l'idée d'un réseau de commutation par paquets n'ont rien à voir avec les préoccupations militaires qui ont poussé Baran. Davies veut simplement créer un nouveau réseau public de communication. Il prévoit 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 épousent les idées de Davies. Cela l'encourage à chercher un financement pour bâtir un réseau expérimental au NPL ([HL–96], pp.78–81).

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

1.2.5 Réseaux locaux et interréseaux

Tandis qu'ARPAnet se propage dans les universités et les organismes de recherche, un autre concept prend forme, celui de **réseau local** (LAN pour *Local Area Network* en anglais), qui est généralement un réseau à une échelle géographique relativement restreinte, par exemple un bâtiment ou un site d'entreprise.

Le pionnier dans ce domaine est le réseau Ethernet conçu au centre Xerox PARC (*Palo Alto Research Center* dans les années 1970, puis IBM lance son propre système, l'anneau à jeton ou *Token Ring*, dans les années 1980.

Lorsqu'on a besoin d'un réseau plus large, on interconnecte plsieurs réseaux locaux pour obtenir un **interréseau** (*internet* en anglais, avec un 'i' minuscule). Dans ce cas une **passerelle** (*gateway* en anglais) permet de relier deux réseaux informatiques (locaux ou interréseaux) de types différents.

1.3 Les logiciels réseau : les protocoles

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

Comme pour les systèmes informatiques[3], la distinction entre *matériel* et *logiciel* est importante dans le cas des réseaux. Nous ne nous intéresserons ici qu'à la partie logicielle des réseaux.

1.3.1 Notion de protocole

Dans un réseau informatique, lorsqu'un ordinateur envoie des informations à un autre, les matériels et les logiciels sont en général de types 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 Marrill 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.

1.3.2 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 de multiples 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.

1.3.3 Systèmes propriétaires et systèmes ouverts

Au début, il n'y a qu'un réseau (ARPAnet) avec ses protocoles associés. Devant le succès de celui-ci, des produits commerciaux sont développés, en particulier pour les réseaux locaux, par exemple reliant les équipements d'une entreprise, et étendus. Chaque constructeur a 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.

2. Désormais nous ne parlerons que de *réseau* (*network* ou même *net* en anglais) 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 numériseur) et logiciels. De nos jours, les outils réseau font partie intégrante du système informatique, dans un sens plus large.

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 concernant les réseaux, mais à propos des systèmes d'exploitation. Jusqu'aux années 1980, chaque constructeur de matériel a sa ligne de produits, et celui qui a fait le choix d'un constructeur est dans une très large mesure dépendant de celui-ci pour tout ce qui concerne le matériel, certes, mais également le logiciel. Certains constructeurs profitent de cette situation pour pratiquer des tarifs prohibitifs ou pour imposer certaines configurations à leurs clients. Le ressentiment des clients prend une telle ampleur que ces derniers finissent par imposer leurs souhaits. DEC (*Digital Equipment Corporation*) passe d'un système d'exploitation propriétaire, VMS (*Virtual Memory System*), sur ses mini-ordinateurs à un système d'exploitation ouvert de type Unix. Ses clients lui en sont gré et l'entreprise vend plus de machines.

1.4 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**.

1.4.1 Étude générale

1.4.1.1 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. Chaque groupe de tâches proches est appelé une **couche**. On obtient alors des **modèles 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.

1.4.1.2 Pile réseau

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

Chaque protocole encapsule les données dans une **unité d'information** plus grande comprenant un **en-tête** et, quelquefois, un **suffixe** :

Supposons, pour simplifier pour l'instant, qu'il n'y ait pas de suffixe. Les données de l'utilisateur sont encapsulées avec l'en-tête du niveau N :

en-tête N	données

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 :

en-tête N-1	en-tête N	données

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

en-tête 1	. . .	en-tête N-1	en-tête N	données

L'intérêt des modèles en couche est, qu'à chaque niveau, on n'a 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).

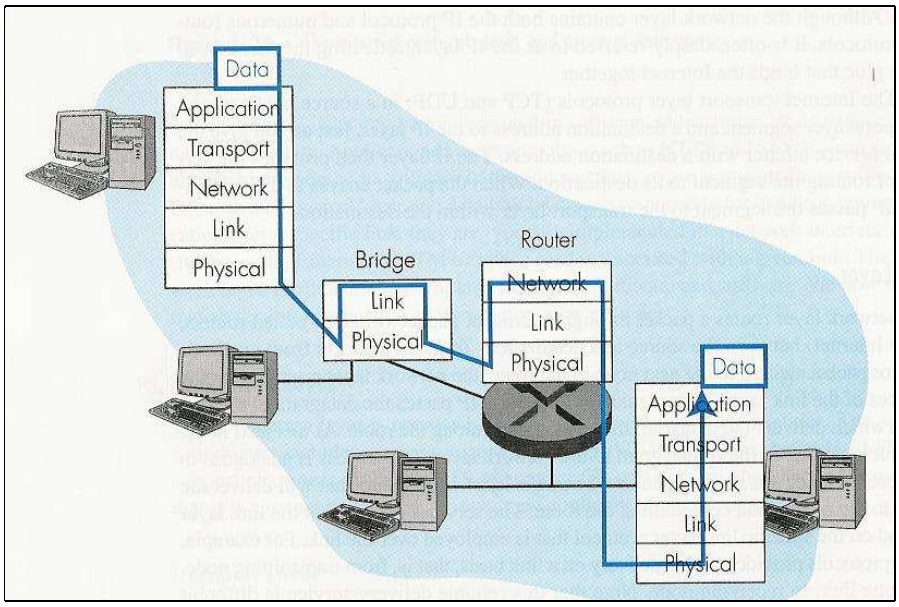


FIGURE 1.4 – Pile réseau

1.4.1.3 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 considère que les données brutes ;
— un *routeur* ne considère que les en-têtes des couches les plus basses pour obtenir l'adresse réseau de destination.

Deux modèles furent développés pratiquement en même temps : le **modèle OSI** (par l'organisme international de normalisation 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, qui demeure en théorie la référence.

1.4.2 Modèle OSI

L'OIN (*Organisation Internationale de Normalisation*, souvent citée sous son acronyme anglais ISO, pour*International Standardisation Organisation*), fondée en 1947, est l'organisation de normalisation la plus prioritaire pour les pays membres (c'est-à-dire pratiquement le monde entier). Elle a proposé un modèle de réseau informatique en sept couches en 1984 [ISO 7498-1], [ISO 7498-2], [ISO 7498-3] et [ISO 7498-4], appelé **modèle OSI**, plus exactement **OSI–RM** pour *Open Systems Interconnection Reference Model* (modèle de référence d'interconnexion des systèmes ouverts) :

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

Ce modèle OSI ne décrit pas l'implémentation réelle d'un système particulier, il se contente de définir les tâches des différentes couches :

— La **couche physique** (*physical layer* en anglais) contrôle la transmission des différents bits *via* un support physique (*medium* 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).

  Elle concerne ce qu'on appelle les réseaux locaux.
— 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**.

Elle concerne ce qu'on appelle les inter-réseaux.

— 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 charger 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.

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. Elles ne font aucune différence entre les diverses applications.

1.4.3 Modèle TCP/IP

L'architecture TCP/IP est analogue 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) permet, comme dans le modèle OSI, le multiplexage et le démultiplexage entre les applications de systèmes d'extrémité.
- La **couche Internet** (*Internet layer* en anglais) est principalement chargée de router les paquets IP de l'expéditeur au destinataire à travers le réseau (c'est-à-dire l'inter-réseau). Elle correspond à la couche réseau du modèle OSI.

1.4.4 Modèle hybride

L'Internet et la plupart des réseaux intranets d'entreprise ont recours à un **modèle 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.

1.5 L'architecture TCP/IP

Nous venons de voir qu'il existe deux grands modèles : OSI et TCP/IP. Ces modèles ne font que définir les fonctionnalités mais n'indiquent rien sur la réalisation de celles-ci. Il existe de nombreuses suites de logiciels réseau, mettant en œuvre tout ou partie de ces modèles. On peut citer, parmi les systèmes propriétaires, **XNS** (pour *Xerox Networking Systems*), **SNA** (déjà cité) et **NetBIOS** d'IBM et, parmi les systèmes ouverts, **UUCP** (pour *Unix-to-Unix Copy Protocol*) et l'architecture TCP/IP (souvent également appelée **architecture Internet** car la suite de protocoles TCP/IP et l'inter-réseau Internet sont étroitement liés).

On peut distinguer le **modèle TCP/IP** (précisant le nombre de couches et le noms de celles-ci), l'**architecture TCP/IP** (qui est une suite de protocoles) et l'**implémentation de TCP/IP** sur tel ou tel système (qui est une suite de logiciels).

1.5.1 Historique

1.5.1.1 Les premiers protocoles d'application

Lors des premières années d'utilisation du premier réseau, l'ARPAnet, de nombreuses – et parfois vigoureuses – discussions portent sur la finalité et l'utilité du réseau, ce qui a pour effet d'affiner et de modifier le logiciel réseau à mesure que les utilisateurs demandent de nouvelles fonctionnalités.

Les deux fonctionnalités les plus demandées sont, au début, la possibilité de transférer des fichiers d'une machine à une autre et la possibilité de se connecter d'un ordinateur à un autre, à distance donc. La connexion à distance permet à un utilisateur situé à Santa Barbara de se connecter, par le biais du réseau, à une machine située à Los Angeles et d'utiliser cette dernière comme s'il se trouvait devant. Les logiciels et protocoles utilisés alors ne sont pas en mesure de gérer ces nouvelles fonctionnalités. Il faut donc continuellement développer, affiner et tester de nouveaux protocoles.

La connexion à distance est finalement implémentée grâce à un protocole appelé **NCP** (dont a déjà parlé) et le transfert de fichiers à distance au moyen du **FTP** (pour *File Transfer Protocol*).

1.5.1.2 Origine de TCP/IP : 1974

Vers 1973, il devient clair que l'ensemble de protocoles utilisés n'est pas capable de gérer le volume de trafic et les nouvelles fonctionnalités requises par les utilisateurs. On s'attelle alors au développement d'une nouvelle suite de protocoles. L'architecture TCP/IP est proposée pour la première fois en 1974 par Vincent Cerf et Kahn [CK–74].

Cerf et Kahn proposent de plus que la nouvelle suite de protocoles soit indépendante du réseau sous-jacent et du matériel informatique. Il s'agit d'une idée audacieuse dans un monde informatique où le logiciel et le matériel sont propriétaires. Cela permet de faire participer n'importe quelle plate-forme au réseau. La suite de protocoles est développée et recevra par la suite le nom de TCP/IP.

1.5.1.3 Envolée : 1982

Une série de RFC, dont nous reparlerons tout au long de ce livre, est proposée en 1981 pour standardiser la version 4 de TCP/IP, plus particulièrement destinée à l'ARPAnet. En 1982, TCP/IP prend la place de NCP comme protocole dominant dans un réseau maintenant étendu, puisqu'il est composé de machines situées un peu partout aux États-Unis. On estime en effet qu'au cours de la première décennie d'existence d'ARPAnet, un IMP y est raccordé tous les 20 jours.

TCP/IP devient important lorsque le Département de la Défense américain (DOD pour *Department Of Defense*) commence à inclure les protocoles de TCP/IP dans les standards militaires, qui sont obligatoires dans de nombreux contrats.

1.5.1.4 L'implémentation BSD : 1983

La définition des protocoles, c'est bien, leur implémentation c'est mieux. L'université de Californie à Berkeley (UCB) reçoit au début des années 1980 une subvention de la DARPA pour qu'elle modifie son système d'exploitation UNIX, connu sous le nom de **BSD** (pour *Berkeley System Distribution*), de manière à ce qu'il inclut la prise en charge d'IP. La version 4.2BSD sort en 1983 avec une implémentation des quatres protocoles TCP, IP, SMTP et ARP (dont nous reparlerons).

Cette version de BSD est mise dans le domaine public. Le succès de 4.2BSD entraîne celui de TCP/IP, lui-même dopé par le développement d'ARPAnet.

La prise en charge d'IP par 4.2BSD est fort bonne mais l'usage en est limité aux seuls petits réseaux locaux. Pour augmenter les capacités de prise en charge d'IP, BSD ajoute des capacités de retransmission, des informations de durée de vie (**TTL** pour *Time To Live*) et des messages de redirection. D'autres fonctions sont également ajoutées pour fonctionner avec des réseaux plus grands, des inter-réseaux et des systèmes étendus connectés par des lignes spécialisées. Ceci donne lieu à une version améliorée (qui contient ce qu'on appelle les **utilitaires de Berkeley**, ou *Berkeley Utilities*) de son système en 1986, sous le nom de 4.3BSD. Une implémentation optimisée de TCP suit en 1988 (4.3BSD/Tahoe). Pratiquement toutes les implémentation de TCP/IP actuellement disponibles plongent leurs racines dans les versions de Berkeley, même si BSD n'a pas connu de nouvelle version depuis 1993.

1.5.2 Définition de standards

1.5.2.1 Les RFC (1969)

Nous avons vu ci-dessus que relier un IMP à l'ARPAnet est l'objet d'un protocole, le protocole 1822. Par contre rien n'est défini, au début, pour savoir comment un utilisateur (**hôte** dans le langage des réseaux) se relie à l'IMP. En 1968 le réseau de l'ARPA commence à se mettre en place ; plusieurs nœuds sont opérationnels et cela commence à se savoir. Au cours de l'été 1968, un petit groupe d'étudiants de second cycle se réunit à Santa Barbara ; ils viennent des quatre sites hôtes – UCLA, le SRI, l'université de Californie à Santa Barbara et l'université de l'Utah. Ils savent que le réseau est en préparation, mais n'ont guère de détails. Il sort de cette réunion un corps de jeunes chercheurs qui vont se consacrer aux communications d'hôte à hôte sur le réseau. Pour activer les choses, ils décident de se rencontrer régulièrement. Un mois après la formation du groupe, il devient évident qu'ils ont tout intérêt à rassembler des notes sur leurs discussions. L'un d'entre eux, CROCKER, propose de rédiger les premiers comptes rendus. Pour ne pas froisser les concepteurs officiels du réseau et pour éviter de paraître trop catégorique, il appelle sa première note *Request for Comments* (ou **RFC**, demande de commentaires), et l'expédie le 7 avril 1969. Intitulée « *le logiciel de l'hôte* » (*Host Software* dans sa version originelle en anglais), la note [RFC 1] est distribuée aux autres sites comme vont l'être les premiers RFC : par la poste, dans une enveloppe en léchant le timbre.

« Demande de commentaires » s'est révélé un choix parfait pour un titre. Cela a l'air à la fois attentif et sérieux. Et cela a tenu. Le langage du RFC est chaleureux et accueillant. L'idée est de favoriser la coopération, sans pontifier. Le fait que CROCKER ait laissé son ego à l'écart lors du premier RFC donne le ton et inspire ceux qui suivent le mouvement dans les centaines de RFC amicaux et serviables qui sont venus après. Les RFC vont devenir le principal moyen de libre expression chez les gens du réseau. Il y a maintenant plus de 3 000 RFC.

1.5.2.2 Le NWG (1969)

Bientôt, le collectif né au cours de l'été 1968 se désigné sous le nom de **Network Working Group** (ou **NWG**, le groupe de travail du réseau). Le défi qu'il faut relever est de trouver un accord de principe sur les protocoles – comment partager les ressources, comment transférer les données, comment faire fonctionner le système. Cela signifie écrire des programmes ou, du moins, adopter certaines règles sur la façon de les écrire, des règles qui recueillent une large adhésion.

L'accord est la condition *sine qua non*. On est en présence d'une communauté fondée sur l'égalité des compétences. Tout le monde peut écrire un code – ou réécrire le code déjà écrit par un autre. Le NWG est une « adhocratie », une aristocratie de spécialistes, une aristocratie égalitaire pour férus d'informatique.

Devançant la construction du réseau, le NWG continue à se réunir régulièrement, et des termes neufs comme des inventions nouvelles sont souvent nés d'un commun accord.

1.5.2.3 Le NIC (1969)

En 1967, lorsque Taylor et Larry Roberts annoncent au colloque d'Ann Arbor leur projet de réseau en douze sites, Doug Engelbart se trouve dans l'auditoire. Il dirige à l'époque un laboratoire de recherches informatiques au SRI. Ce qui l'intéresse, c'est l'emploi de l'ordinateur pour élargir l'intellect humain. Sous contrat avec l'ARPA, il s'occupe de mettre au point un système appelé **NLS** (pour *oNLine System*, système en ligne, premier essai d'hypertexte). Engelbart considère le NLS comme l'instrument naturel d'un bureau central de renseignements pour le réseau de l'ARPA. Si on doit partager des ressources, il est important de faire savoir à chacun ce qui est disponible. À la réunion du Michigan, il offre de monter le centre d'information du réseau, le *Network Information Center*, lequel finit par être connu sous le nom de **NIC** ([HL–96], pp. 93–94).

Le SRI suppporte financièrement le NIC financièrement, tenu par Elizabeth (Jake) Feinler, qui maintient des tables des noms d'hôtes et de leurs adresses et un répertoire des RFC.

1.5.2.4 Création d'organismes de contrôle

Lorsque le DARPA est créé en 1980, un groupe est formé pour développer un ensemble de standards pour Internet. Ce groupe, appelé **Internet Configuration Control Board** (**ICCB**) est réorganisé en 1983, date à laquelle il devient l'**Internet Activities Board** (**IAB**), dont la tâche est de concevoir, de mettre en œuvre et de gérer **Internet**, désormais nom officiel du réseau ARPAnet.

En 1986, l'IAB se décharge du développement des standards sur l'**Internet Engineering Task Force** (**IETF**) et la recherche à long terme est confiée à l'**Internet Research Task Force** (**IRTF**). L'IAB se réserve le droit de donner ou non son autorisation à tout ce que peuvent proposer les deux organisations qui dépendent de lui.

La dernière étape de cette saga est la formation de l'**Internet Society** en 1992, date à laquelle l'IAB devient l'**Internet Architecture Board**. Ce groupe reste responsable des standards existants et à venir, et soumet ses travaux à la direction de l'*Internet Society*.

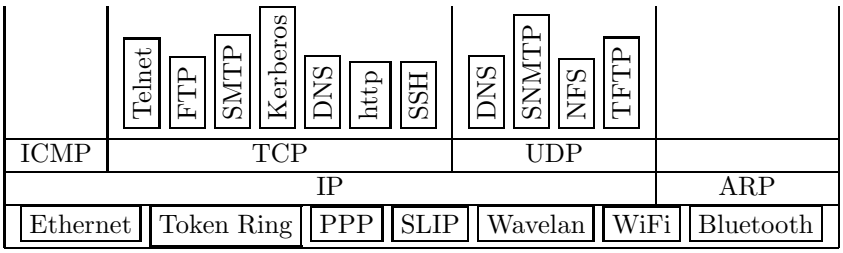
Pratiquement dès ses origines, Internet est défini comme une « collaboration internationale à organisation souple de réseaux autonomes et interconnectés », qui prend en charge des communications d'hôte à hôte « par le biais de l'adoption volontaire de protocoles et procédures ouverts » définis dans un document technique appelé *Internet Standards*, RFC 1310,2 ([RFC 1310]).

L'IETF continue à affiner les standards de communication sur Internet par le biais de divers groupes de travail, chacun spécialisé dans un aspect du protocole. Certains groupes sont spécialisés dans la gestion des réseaux, d'autres dans la sécurité, les services utilisateurs ou le routage. La

plupart du temps, les groupes de l'IETF se forment, créent une recommandation et se dissolvent en moins d'un an.

1.6 Les protocoles de l'architecture TCP/IP

La figure suivante montre les protocoles de base de l'architecture TCP/IP :



1.6.1 Protocoles de la couche d'accès

L'architecture TCP/IP proprement dite ne s'occupe de l'analogue ni de la couche physique, ni de la couche liaison du modèle OSI (bien que l'IETF ait développé un protocole tel que PPP). Les cartes réseau et leurs pilotes se trouvent sous la couche Internet, que l'on appelle alors quelquefois **couche d'accès** pour TCP/IP au sens étendu.

On y trouve les protocoles **Ethernet** (standard Xeros puis IEEE), **Token Ring** (ou *anneau à jeton*, standard IBM puis IEEE), les bus à jeton (*token bus*), **PPP** (pour *Point-to-Point Protocol*, [RFC 1661]), **SLIP** (pour *Serial Line Internet Protocol*, [RFC 1055], liaison série par modem ou par RNIS, *Réseau Numérique à Intégration de Services*, appelé *numeris* en France), **ATM** (pour *Asynchronous Transfer Mode*, dont nous avons déjà parlé), **X.25** (norme de l'**UIT**, *Union Internationale des Télécommunications*), **Frame Relay**, **WiFi** (standard IEEE de *Wireless Fidelity*), **FTTH** (pour *Fiber to the Home*, soit « Fibre optique jusqu'au domicile ») et **Bluetooth** (standard IEEE).

Chacun de ces protocoles possède une partie qui dépend du support, et donc de la couche physique. Par exemple, Ethernet possède des protocoles pour les paires torsadées, pour les câbles coaxiaux, pour les fibres optiques, et ainsi de suite.

1.6.2 Protocoles de la couche réseau

La couche réseau comprend trois protocoles principaux :

— Le protocole principal s'appelle **IP** (pour *Internet Protocol*, [RFC 791]). Il est chargé de déplacer sur les réseaux les paquets assemblés par les protocoles de la couche du dessus. Il utilise un ensemble d'adresses uniques, appelées **adresses IP**, pour chaque composant du réseau afin de déterminer le routage et les destinations.

  En fait IP se décline en plusieurs versions dont trois sont actuellement utilisées :

  — la version 4, **IPv4**, avec ses adresses IP sur 32 bits ;

  — la version 6, **IPv6** ou **IPng** (pour *IP New Generation*), [RFC 2460], avec ses adresses IP sur 128 bits ;

  — la version sécurisée, **IPsec** [RFC 2401], qui n'envoie pas les informations en clair.

  Par défaut, IP signifie IPv4.

— Le protocole **ICMP** (pour *Internet Control Message Protocol*, [RFC 792]) est chargé de vérifier l'état des composants sur un réseau et de générer des messages sur ces états. Il

peut être utilisé pour informer d'autres composants du dysfonctionnement d'une machine donnée. Lorsque IP ne peut pas transmettre un paquet à destination, il charge ICMP d'en avertir l'expéditeur et en reste là (la couche supérieure décide s'il faut renvoyer le paquet ou non).

Bien que considéré comme un protocole de la couche réseau, ICMP repose sur IP dans la mesure où ses messages sont envoyés *via* IP ; c'est pourquoi il apparaît au niveau de la couche transport sur la figure ci-dessus.

— **ARP** (pour *Address Resolution Protocol*, [RFC 826]) est installé entre la couche d'accès et la couche Internet. Ce protocole est mis en place pour traduire les adresses réseau (les adresses IP par exemple) en adresses physiques.

1.6.3 Protocoles de la couche de transport

La couche de transport comprend deux protocoles fondamentaux :

— **TCP** (pour *Transmission Control Protocol*, [RFC 793]) est celui qui fait au mieux pour permettre un transfert de données fiable.

— **UDP** (pour *User Datagram Protocol*, [RFC 768]) n'assure pas la retransmission des datagrammes si ceux-ci n'arrivent pas à destination. Le composant émetteur n'a même aucun moyen de savoir si un message a été reçu correctement ou non. UDP est donc moins fiable que TCP mais les datagrammes sont moins lourds.

On dit que UDP est **orienté sans connexion** (*connectionless* en anglais) alors que TCP est **orienté connexion**.

1.6.4 Les protocoles d'application

Les applications s'appuient sur TCP (telnet ou FTP, par exemple), sur UDP (TFTP et NFS, par exemple) ou peuvent reposer sur l'un des deux protocoles de la couche de transport au choix (DNS, par exemple) :

— Les **connexions à distance** (*remote connection* en anglais) permettent à un utilisateur basé sur un système de se connecter – par l'intermédiaire du réseau – à un autre système l'acceptant comme utilisateur. On parle de **connexion distribuée**, ce qui est différent, lorsque, comme dans le système SAGE, le terminal est situé à une certaine distance mais les systèmes sont les mêmes. Le protocole `telnet` [RFC 854] est le premier à permettre la connexion à distance sous TCP/IP.

En raison de la transmission sans garantie du mot de passe et des données, on préfère aujourd'hui le protocole **SSH** (*Secure Socket Shell*, [RFC 4251]) à `telnet`.

— Les **transferts de fichiers** permettent aux utilisateurs de partager des fichiers rapidement et efficacement, sans duplication excessive ni nécessité de se préoccuper de la méthode de transport. Il est bien plus rapide de transférer un fichier par réseau que par la poste, et même que de copier le fichier sur une disquette (ou une clé USB) pour le porter d'une pièce à une autre. **FTP** (pour *File Transfer Protocol*, [RFC 765, RFC 959]) est le premier protocole utilisé sous TCP/IP : il date de 1971 [RFC 172].

Ayant l'inconvénient de transmettre le mot de passe en clair, il est de plus en plus fréquemment remplacé par **SCP** (pour *Secure Copy Protocol*) et **SFTP** (pour *Secure File Transfer Protocol*), faisant partie de la même suite de protocoles que SSH.

— Le **courrier électronique** est bon marché (pas d'enveloppe, ni papier, ni timbre) et rapide (le tour du monde en une minute ou presque). Le premier protocole de transfert depuis l'expéditeur jusqu'à la boîte aux lettres est **SMTP** (pour *Simple Mail Transfer Protocol*,

[RFC 822, RFC 2822]), totalement transparent pour l'utilisateur. En coulisses, SMTP se connecte à une machine distante et transfère les messages électroniques. Le protocole date de 1973 [RFC 561].

Le protocole de transfert de la boîte aux lettres vers le destinataire, lorsque celui-ci veut consulter son courrier, est **POP** (*Post Office Protocol*, [RFC 1939]), de plus en plus remplacé par **IMAP** (*Internet Message Access Protocol*).

— Au fil du temps, d'autres protocoles s'y sont ajoutés, comme **DNS** (pour *Domain Name System*, [RFC 1035]) qui convertit les noms de domaine (tels que `www.linux-france.org`) en adresses IP et *vice-versa*.

— **HTTP** (pour *HyperText Transfer Protocol*, [RFC 1945, RFC 2616]) est le protocole le plus fréquemment utilisé actuellement dans la couche application. Il permet l'échange de données dans le **World Wide Web**, c'est-à-dire le chargement de pages web par l'intermédiaire d'un **navigateur** (*browser* en anglais : Netscape, Mozilla, Lynx, etc. puis *Internet Explorer*, *Firefox*, etc.).

Le format des pages Web est l'objet d'un autre protocole : **HTML** pour *HyperText Markup Language*, langage de balisage hypertexte.

— **NFS** (pour *Network File System*, [RFC 1094]) permet à plusieurs ordinateurs d'accéder à un seul et même système de fichiers.