# Modems-HOWTO

Jean Michel VANSTEENE <vanstee@worldnet.net> 13 Février 1996

Le modem est devenu aujourd'hui un produit à la mode. Que ce soit pour l'accès à Internet ou pour se connecter chez un particulier qui laisse bénévolement sa machine disponible, il faut un modem. Or, cet appareil, d'apparence fort simple, cache des choses très sophistiquées et son emploi peut engendrer bien des soucis. J'ai constaté d'ailleurs que bon nombre d'utilisateurs se posaient des questions à son propos. (Et ceux qui ne s'en posent pas ont parfois des problèmes qu'ils seraient aptes à résoudre par eux-même, s'ils connaissaient un peu son fonctionnement.) Ce document n'est ni un HOWTO ni une FAQ, ce qui est à priori inhabituel pour un document Linux. J'ai longuement réfléchi avant de me lancer dans cette aventure. Parmi mes priorités, la première a été d'être clair et de ne pas tomber dans le genre cours magistral... Le but de ce document est en fait d'éclaircir un peu des notions *dont on a entendu parler* : bande passante, bits/seconde, baud, modulation, interface série, connexions à vingt-huit-huit ... Après avoir lu ce document, de deux choses l'une : soit vous vous dites *c'est imbitable*, et là vous sautez sur votre courrier-é préféré pour m'engueu... me le dire, soit vous pensez que ca vous a apporté quelque-chose et là, vous sautez sur votre courrier-é... pour me le dire. En tout cas, toute remarque sera la bienvenue, comme d'habitude.

## Contents

**1 Introduction** **3**

1.1 Résumons un peu . . . 4

**2 Un peu de théorie du signal...** **4**

2.1 Les supports de transmission . . . 4

**3 Les télécommunications analogiques et numériques** **4**

3.1 Les télécommunications analogiques . . . 4

3.2 Le signal téléphonique . . . 5

3.3 Les télécommunications numériques . . . 5

3.4 Alors pourquoi l'analogique ? . . . 5

3.5 De l'analogique au numérique et réciproquement . . . 5

3.5.1 De l'analogique au numérique . . . 6

3.5.2 Du numérique à l'analogique . . . 6

3.6 Des bits et des débits . . . 6

3.6.1 Le débit binaire maximum . . . 6

3.6.2 La capacité de transmission maximale . . . 7

**4 Le modem** **7**

4.1 Vocabulaire . . . 8

4.2 Le mode de transmission . . . 8

4.3 Le support de transmission ou ligne . . . 8

4.4 L'adaptation du signal . . . 9
4.4.1 Exemple . . . 9
4.4.2 Résumé . . . 10
4.5 Le dialogue . . . 10
4.5.1 La jonction série . . . 10
4.5.2 Le dialogue proprement dit . . . 11
4.5.3 Le contrôle de flux . . . 11
4.6 La connexion au réseau téléphonique commuté . . . 12
4.6.1 Initialisation du modem . . . 12
4.6.2 Établissement de la connexion . . . 12
4.6.3 Réponse automatique ou manuelle . . . 13
4.6.4 Déconnexion . . . 14

**5 Etat actuel de la normalisation 14**
5.1 À propos du V.42 bis . . . 15
5.2 Débits et modulations . . . 15

**6 Foire Aux Questions 17**

**7 Un mot sur les empilements protocolaires couramment utilisés 17**
7.1 TCP/UDP/IP . . . 18
7.2 PPP/SLIP . . . 18
7.3 Mise en oeuvre . . . 19
7.4 Les fichiers de configuration . . . 20
7.4.1 Les applications . . . 20
7.4.2 Les couches de communication : TCP/UDP/IP . . . 20

**8 Le Minitel 20**
8.1 L'écran . . . 21
8.2 Le clavier . . . 21
8.3 Le modem . . . 22
8.4 Utilisation du Minitel comme simple terminal . . . 22
8.4.1 Configuration de getty . . . 22
8.5 Utilisation du Minitel pour un accès distant . . . 23

**9 Un modem particulier: le câble null-modem 23**

**10 Choix d'un modem 24**

**11 Quelques chiffres** **25**

11.1 Débit réel . . . . . . . . . . 25

11.2 Coûts de connexion . . . . . . . . . . 25

**12 Envisager d'écrire des applications** **26**

12.1 Et si c'était simple ? . . . . . . . . . . 26

12.1.1 open, close . . . . . . . . . . 26

12.1.2 read, write . . . . . . . . . . 26

12.1.3 ioctl . . . . . . . . . . 26

12.2 Appels entrants (Dial-in) et appels sortants (Call-out) sous Linux . . . . . . . . . . 27

12.2.1 Introduction . . . . . . . . . . 27

12.2.2 Gestion . . . . . . . . . . 27

**13 Quelques applications intéressantes** **28**

13.1 Un numéroteur . . . . . . . . . . 28

13.2 modemstat et compagnie . . . . . . . . . . 29

13.3 Un détecteur de signal 109 (CD) . . . . . . . . . . 29

**14 Remerciements** **30**

# 1 Introduction

La communication a toujours été, est et sera toujours un échange de signaux entre un émetteur et un récepteur. Afin d'avoir les idées claires sur ce que nous allons aborder, décomposons les différentes étapes de la communication. Le meilleur modèle que nous allons prendre est l'homme qui l'utilise depuis fort longtemps.

**Première étape** : prenons un homme, bien rasé de préférence, propre et prêt à se rendre au travail. Justement il a un mot à dire à sa femme avant de partir. C'est l'information à transmettre.

**Deuxième étape** : comment la transmettre. Si elle est là, il crie (bon, il parle), sinon il écrit le message sur un bout de papier. Notre homme est donc capable (voyez-vous ça, il est à peine 7h30 du matin !) de coder son information en fonction de la manière dont il transmet son message.

Pour communiquer, nous utilisons des éléments de base dont l'ensemble forme l'alphabet. Une succession de ces éléments définit un vocabulaire. En fait il s'agit ni plus ni moins que d'un code, complexe certes, mais compréhensible par tous ceux qui l'adoptent. Moins il est ambigu, plus il est précis. (Vous pouvez essayer de donner trois sens différents à cette phrase pour comprendre que notre langue est parfois ambiguë : il est énormément bête .) Pour s'exprimer, il est ensuite capable de découper une suite de mots (éléments continus) en phonèmes (éléments discontinus), que le récepteur saura réassembler.

En informatique, l'information de base est **binaire**, donc codée sur deux valeurs logiques que l'on note habituellement 0 et 1. C'est le code sans doute le plus élémentaire qui soit. Aussi il existe un certain nombre de codes intermédiaires. Nous citerons par exemple le code ASCII, permettant de coder les lettres et chiffres.

**Troisième étape** : sa femme découvre le message (ou l'entend). Elle est capable de le reconstituer. Les lettres (respectivement les phonèmes) forment des mots qui forment des phrases qui forment le message. Ouf ! On y est.

### 1.1 Résumons un peu

**Découpage horizontal**. La communication n'est possible que s'il existe un code commun. À tout niveau il faut s'assurer non seulement que le code employé a un sens, mais en plus qu'il a le même pour l'émetteur et le récepteur. On parle alors de *protocole*. Au niveau le plus bas, un signal est utilisé comme un moyen de communication. Il transporte en effet un message sous une forme particulière appelé *codage* ou *modulation*. Un signal a une nature physique et un modèle mathématique. Nous nous étendrons davantage sur sa nature que sur le modèle qui, bien qu'intéressant, nous amènerait trop loin. Le signal s'appuie sur un support.

**Découpage vertical**. De l'idée au code employé : plusieurs niveaux de traitement sont utilisés pour transformer un message complexe en éléments plus simples aptes à être véhiculés et compris par une entité homologue.

Or s'il y a un signal, il faut forcément un support de transmission, permettant de le véhiculer d'un point à un autre. Nous verrons cela un peu plus loin. Celui qui nous intéresse concerne les transmissions téléphoniques.

Voici donc jetées les bases de la communication. Nous allons maintenant éclaircir un peu tout cela dans les différentes parties qui vont suivre. La première étape consiste à consolider les bases sur les signaux, ensuite nous verrons leur transmission.

## 2 Un peu de théorie du signal...

La voix est un bon exemple de signal permettant de véhiculer une information. Ce signal est caractérisé par sa **bande passante**, c'est-à-dire le domaine de fréquences sur lequel elle peut s'étendre. En général cette bande est continue et comprise entre 30 et 15000 Hz. Ce signal est de type sinusoïdal.

Sans entrer dans des détails mathématiques, disons qu'un signal est composé d'une fréquence principale et d'harmoniques. Il est possible d'en donner une représentation mathématique grâce aux séries de Fourier, mais nous n'irons pas plus loin. Disons simplement que ce signal est appelé signal "analogique", parce qu'il peut prendre n'importe quelle valeur de façon continue entre deux instants : le signal est "modulé".

### 2.1 Les supports de transmission

Un signal quel qu'il soit, n'a d'intérêt que s'il peut être transporté. Il faut savoir qu'un système de transmission n'est jamais en mesure d'émettre des signaux sans leur faire subir de déformations : selon leur nature, on parle de distorsion, d'affaiblissement, de diaphonie ... Comme nous le verrons plus loin, les lignes téléphoniques ne font pas exception à cette règle.

Chaque type de support est caractérisé entre autres par son aptitude à transmettre un signal plus ou moins fidèlement. De nombreux supports sont utilisés en transmission de données : les supports avec guide physique (câbles, fibres, ...) et les supports sans guide physique (ondes radio, ondes lumineuses). Pour donner une idée, de la qualité des supports, disons que les câbles électriques à paires torsadées sont les moins fiables, suivis par les câbles coaxiaux. Les fibres optiques offrent actuellement le meilleur compromis fiabilité/performance.

## 3 Les télécommunications analogiques et numériques

### 3.1 Les télécommunications analogiques

C'est un mode de communication utilisé depuis très longtemps notamment dans la technologie téléphonique. Il s'agit en effet d'une activité beaucoup moins consommatrice de ressources, tant financières que technologiques que la transmission numérique. On n'est pas tout à fait prêt à pouvoir s'en passer.

## 3.2 Le signal téléphonique

À l'origine, le téléphone a été conçu pour transmettre la voix. Malheureusement, il n'est pas possible, étant donné le support utilisé, de véhiculer le signal complet, c'est-à-dire l'ensemble des fréquences le constituant. Le domaine de fréquences (on parle de largeur de bande) que peuvent transmettre les lignes téléphoniques est officiellement compris entre 300 et 3400 hertz[1]. On applique donc au signal de départ un *filtre passe-bande* qui restreint l'espace de fréquence attribué à la transmission du signal sur cette liaison. Ceci correspond heureusement à 90% de netteté de la voix.

Selon le principe bien admis que tout traitement a un coût, le plus simple et le moins coûteux en télécommunications est de transmettre le signal avec le moins de transformations possible. C'est bien ce qui se passe pour la voix par téléphone. Les seules transformations sont d'ordre analogique comme l'amplification par exemple.

## 3.3 Les télécommunications numériques

Nous avons déjà évoqué précédemment que le fonctionnement de nos chers ordinateurs était fondé sur la seule information binaire. Celle-ci est représentée, dès lors qu'il s'agit de la visualiser (oscilloscope) ou de la transporter, par un signal rectangulaire à deux niveaux.

Pour transporter un tel signal, le plus simple et le moins coûteux consiste à lui faire subir le moins de traitement possible, voire à le transporter tel quel. On imagine aisément que pour transmettre ce signal sur un support, il suffise de définir deux signaux électriques représentant les niveaux logiques 0 et 1. De plus ce type de transmission offre des performances considérablement supérieures à la transmission analogique, ceci pour deux raisons.

La première est un faible taux d'erreurs. En effet, alors qu'en transmission numérique, les signaux sont transmis avec des tensions d'amplitude variable, en transmission numérique le nombre de niveaux est limité. Les signaux parasites s'infiltrant dans un signal analogique sont donc très difficiles à supprimer et engendrent des erreurs. En transmission numérique, les défauts sont plus facilement repérables et les équipements régénèrent plus facilement un signal parasité ou affaibli.

La deuxième raison tient au fait que l'on sait mieux traiter une information numérique. Ainsi, en utilisant les méthodes de multiplexage, de compression, l'acheminement des données se fait beaucoup plus rapidement. De plus le coût du matériel de traitement diminue considérablement.

## 3.4 Alors pourquoi l'analogique ?

Cette question est bien entendu la première que l'on se pose maintenant. La réponse tient en quelques mots : essentiellement pour des raisons financières. Lorsque les ordinateurs sont organisés en petits groupes fermés, l'infrastructure à mettre en place pour les relier est bien sûr numérique. Mais dès lors que les communications s'établissent sur de grandes distances, elles doivent empreinter l'infrastructure existante, qui est analogique. L'évolution se fait lentement vers le tout numérique, Numéris en est l'exemple prometteur.

## 3.5 De l'analogique au numérique et réciproquement

Puisqu'il faut s'adapter à un monde fait de numérique et d'analogique, il faut savoir passer de l'un à l'autre. C'est essentiellement ce qui va se passer avec les modems. Faisons d'abord un point rapide sur les méthodes de conversion entre l'analogique et le numérique.

[1]Les *codecs* (codeurs-décodeurs) modernes utilisés dans les centraux téléphoniques actuels ont une bande passante de l'ordre de 200 à 3700 Hz et la qualité des lignes des abonnés s'en trouve généralement améliorée.

### 3.5.1 De l'analogique au numérique

L'information de départ est représentée par un signal qui, si on le transforme en tensions électriques, peut prendre une infinité de valeurs (dans un intervalle fini, heureusement !) entre deux instants. Pour le transcrire dans un monde fait d'un nombre limité de niveaux significatifs, il faut le coder. Un des principes de codage les plus simples consiste à prélever à intervalle régulier la valeur de la tension, puis de la représenter en binaire sur 7 ou 8 bits. Le prélèvement est usuellement appelé *échantillonnage* et la fréquence d'échantillonnage correspond au nombre d'échantillons prélevés par seconde. Un codeur-décodeur prélève en général 8000 échantillons par seconde.

### 3.5.2 Du numérique à l'analogique

A l'inverse, la transformation d'une information numérique en analogique consiste à moduler un signal de base en fonction de cette information. C'est le rôle du modulateur-démodulateur (modem).

Considérons maintenant ce signal de base. S'agissant d'un signal analogique, c'est donc une sinusoïde dont la fréquence peut varier, dans le cas qui nous intéresse, de 1000 à 2000 hertz. C'est la porteuse. La modulation de ce signal va consister ensuite à en faire varier un ou plusieurs paramètres : la **phase**, **l'amplitude** ou la **fréquence**.

La modulation d'amplitude consiste à modifier l'amplitude de la porteuse, selon l'information binaire à transmettre. Par exemple une valeur de l'amplitude est attribuée au 0 et une autre au 1.

La modulation de fréquence correspond à la même notion, mais ici les deux valeurs sont représentées par des fréquences différentes.

Enfin, la modulation de phase, consiste à faire varier la phase de la porteuse, de 45, 135, 225 ou 315 degrés par exemple.

La **rapidité de modulation** caractérise la vitesse à laquelle ces changements s'effectuent. C'est la caractéristique essentielle qui permet de définir la bande passante.

Arrêtons-nous là un instant pour évoquer maintenant la notion de débit. Il est en effet facile d'imaginer pouvoir faire varier un signal à volonté, mais ce serait ne pas tenir compte de certaines caractéristiques physiques des supports qui nous contraignent fortement.

## 3.6 Des bits et des débits

Une des valeurs caractéristique des supports de transmission est le débit maximum qu'ils peuvent supporter. Comment s'empêcher de comparer un support à une route. Le nombre maximum de véhicules qu'une autoroute est capable de supporter par heure est très supérieur à celui d'une route départementale (même si vous n'aimez pas les routes départementales, mais ceci est une autre histoire ...).

En ce qui concerne les supports de transmission, leur débit maximum est directement lié à la largeur de la bande passante. Chose promise, chose due, pas trop de mathématiques ici. Mais il est impossible de ne pas parler de deux valeurs fondamentales qui vont permettre de comprendre ce qui se passe avec les modems : ce sont le **débit binaire maximum** et la **capacité de transmission maximale**.

### 3.6.1 Le débit binaire maximum

Sur un canal de transmission dont la bande passante est B, il est montré qu'un signal peut être entièrement reconstitué à l'arrivée, si on le transmet en prenant 2B échantillons par seconde. Le débit maximum s'écrit alors :

```
Dmax = 2B
```

Si, de plus, le signal peut prendre plus de deux valeurs significatives, la formule se généralise en :

```
Dmax = 2B log  V
             2
```

où **V** correspond au nombre de niveaux significatifs (ou états) que peut prendre le signal : c'est sa **valence**. Par exemple, V=4 si le signal peut prendre les valeurs +10 volts, +5 volts, -5 volts et -10 volts.

Ceci pour vous montrer qu'en théorie, sur une ligne téléphonique dont la bande passante est de 3000 hertz, le débit maximum est de 6000 bits/s avec deux niveaux significatifs (un pour le 0, un pour le 1), 12000 bits/s avec quatre niveaux, etc. Le débit maximum est théoriquement infini.

### 3.6.2 La capacité de transmission maximale

Un des inconvénients supplémentaires des supports est le **bruit**. Or la quantité de bruit présente sur une ligne s'exprime par rapport à la puissance utile du signal transmis : c'est le **rapport signal/bruit**. Plus ce rapport est grand, meilleure est la qualité. La capacité de transmission maximale est une fonction de ce rapport. Pour une ligne téléphonique, cette capacité maximale atteint 30000 bits/s. Cela signifie bien que sur ces lignes **on ne peut transmettre à plus de 30000 bits/s**[2] quels que soient la valence et la fréquence du signal. C'est une limite au débit binaire maximum.

# 4 Le modem

Le rôle du modem est d'adapter les signaux rectangulaires de données, que le réseau téléphonique ne peut pas transmettre tels quels, en signaux transmissibles par ce réseau.

Il a en fait deux fonctions :

- un rôle d'**adaptation du signal** aux lignes du réseau utilisé, c'est-à-dire de modulation et de démodulation ;
- un rôle de **dialogue** avec l'équipement informatique auquel il est relié.

Il tient donc exactement le même rôle fonctionnel qu'une couche de communication (TCP, par exemple). Il possède une **interface** permettant un dialogue avec un utilisateur se trouvant à un niveau supérieur. Ici il s'agit d'une interface physique (y compris électrique). Il communique avec une entité paire (un autre modem) selon un **protocole**.

La structure interne d'un modem est décrite ci-dessous :

```
+-----+      +--------+         +-------------+       +---------------+
|  J  +----->| codeur +-------> |  modulateur +------>|               |
|  O  |      +--------+         +-------------+       |               |
|  N  |                                               |               |
|  C  |                                               | transformateur |
|  T  |                                               |               |
```

[2]C'est bien une capacité maximale physique, à ne pas confondre avec des débits logiques après compression de données.

```
|  I  |                                           |     ligne      |
|  O  |      +----------+      +--------------+      |                |
|  N  |<-----+ decodeur |<------+ demodulateur |<-----|                |
+-----+      +----------+      +--------------+      +----------------+
```

Les paramètres caractérisant un modem sont :

- le *débit d'information* en bits/s ;
- le *mode de transmission* : synchrone ou asynchrone ;
- le *support de transmission utilisé* : réseau ou ligne spécialisée ;
- le *mode de couplage* à la ligne : électrique ou acoustique.

Nous aborderons assez rapidement l'ensemble de ces paramètres, selon l'utilisation que nous aurons à en faire. La notion de débit devrait maintenant être assimilée.

Penchons-nous rapidement sur les modes et les supports de transmission utilisés. Voyons ensuite plus précisément le rôle d'adaptation du signal du modem, puis le dialogue qui met en jeu la jonction et la ligne.

Commençons par définir un vocabulaire commun.

## 4.1 Vocabulaire

Un **avis** est une recommandation édictée par l'U.I.T (Union Internationale des Télécommunications), organisation intergouvernementale compétente en télécommunications. Les avis ont valeur de norme au sein de l'Europe, puisque les organismes de Télécom nationaux ont encore le monopole. Les recommandations sont issues de travaux de diverses commissions d'études et sont adoptées lors des assemblées pleinières (délai de l'ordre de neuf mois, étant donné l'évolution rapide des technologies). La section 5 (Etat actuel de la normalisation) décrit les différents avis actuellement en vigueur.

Dans sa normalisation, l'U.I.T définit l'équipement informatique comme un **ETTD** (*Équipement Terminal de Traitement de Données*) et le modem comme un **ETCD** (*Équipement Terminal de Circuit de Données*). La connexion d'un équipement informatique à un modem, par exemple, est réalisée par l'intermédiaire d'une **jonction** ou **interface**.

On appelle half-duplex (bidirectionnel à l'alternat), une transmission s'effectuant dans un seul sens à la fois. On appelle full duplex (bidirectionnel simultané), une transmission pouvant s'effectuer dans les deux sens en même temps. Ces transmissions peuvent avoir lieu indifféremment sur liaison 2 ou 4 fils.

## 4.2 Le mode de transmission

Une transmission de donnée est toujours liée au facteur temps. Dans les transmissions en série qui constituent la majorité des transmissions, l'émetteur et le récepteur doivent travailler à la même cadence. Dans le mode **synchrone**, ils sont calés sur le même rythme grâce à des signaux d'horloge émis avant la transmission. Dans le mode **asynchrone**, l'horloge du récepteur n'est déclenchée puis arrêtée que sur réception de bits de début et de fin. On les appelle bits de **start** et de **stop**. Ce mode, bien que moins performant, est le plus utilisé actuellement dans les communications à travers le réseau public.

## 4.3 Le support de transmission ou ligne

Un modem est utilisable principalement sur deux types de supports : le **réseau commuté** ou la **ligne spécialisée**. Sur chaque type de support, les liaisons peuvent être à deux ou quatre fils.

Dans le cas qui nous intéresse, le modem est relié au réseau téléphonique commuté et la liaison est à deux fils. Nous l'utilisons soit en half duplex, soit en full duplex selon l'avis (voir définition de ce mot au paragraphe 4.1 (Vocabulaire).

A ce propos, réfléchissons un peu sur l'utilisation qui est faite actuellement du Réseau Téléphonique Commuté (appelé aussi RTC). Nous l'utilisons bien souvent en full duplex sur liaison deux fils (avis V.32 ou V.34). Lorsqu'on utilise des lignes à quatre fils, il est facile d'imaginer que l'on consacre deux fils à chaque sens de transmission. Pour chaque sens, considérant les vitesses de modulations maximales possibles, on conçoit qu'il faut combiner plusieurs types de modulations pour obtenir des débits maintenant courants de l'ordre de 28800 bits/s.

Or le RTC n'utilise que **deux** fils. Pour travailler en full-duplex à des débits relativement faibles (en fait jusqu'à l'avis V22 bis), il était possible de partager la bande de fréquence en deux moitiés, une pour chaque sens. Avec les débits employés actuellement ce n'est plus possible. Pour travailler en full-duplex à d'importants débits, il est fait appel à des algorithmes complexes dits “de suppression d'écho” (proche et lointain). Imaginez le travail à réaliser : chaque modem reçoit les données envoyées par le distant mélangées à ses propres données. Le tout est encore pollué par de l'écho ! Et pour compliquer le tout, tout ceci varie dans le temps, et bien sûr d'une communication à l'autre.

Vous comprendrez donc qu'avec une telle utilisation du RTC, les modems soient continuellement soumis à rude épreuve pour ce qui est de la correction, ceci pouvant conduire à des débits variables selon le moment.

## 4.4 L'adaptation du signal

Nous avons vu aux sections précédentes ce qu'étaient une modulation et un débit. Rassemblons maintenant un peu toutes ces idées. Bien souvent, c'est sur ce point délicat que les esprits se perdent. Nous avons vu que la rapidité de modulation est une caractéristique essentielle de la bande passante. Plus cette rapidité est grande, plus la bande passante demandée est large. Sur le réseau téléphonique, la bande maximale officielle est de 3100 hertz (300 à 3400 Hz). Dans les centraux téléphoniques modernes, elle va jusqu'à 3500 Hz.

Pour bien comprendre le mécanisme de l'adaptation du signal, imaginez maintenant que nous disposions d'un appareil électrique capable d'émettre quatre niveaux de tensions possibles.

Les données à transmettre sont quant à elles toujours présentées sous forme d'un flot ininterrompu (ou presque) d'informations binaires.

L'idée serait de regrouper les bits deux par deux et de les faire passer par ce dispositif, afin d'obtenir en sortie le niveau de tension correspondant. Un tel signal en sortie est dit de **valence** 4. Plus généralement, la valence d'un signal est le nombre d'états qu'il peut prendre. Cette transformation du signal est appelée **codage**.

Afin d'adapter ce signal de sortie au support, il faut maintenant le moduler, par exemple en choisissant d'effectuer une modulation de phase. Etant donnée sa valence, nous avons besoin de quatre décalages de phase.

A chaque fois que **deux** bits se présentent, il est possible d'effectuer **une** modulation. A l'autre bout, l'équipement est capable de regénérer deux bits. Le débit (en **bits/s**) est donc bien double de la vitesse de modulation (exprimée en **bauds**).

### 4.4.1 Exemple

Vous configurez un modem à 4800 bits par seconde (V.27 ter). Que va-t-il se passer ? Selon cette norme, le modem va réaliser une modulation de phase différentielle octovalente. Il va donc regrouper les bits par trois (**tribits**) pour moduler le signal. La vitesse de modulation est donc de 1600 bauds et le débit de 4800 bits/seconde. Pour obtenir un débit de 9600 bits par seconde, il faudra combiner un autre type de modulation. La section 5.2 (Débits et modulations) présente l'essentiel des modulations utilisées dans les différentes normes actuelles.

#### 4.4.2 Résumé

L'adaptation du signal peut se faire de trois manières :

- par une simple modulation appropriée ;
- par un codage puis une modulation ;
- par un simple codage. Ce type d'adaptation est présent dans certains modems dits "bande de base" qui transmettent directement ce code sur la ligne. Ce ne sont pas ceux que nous utilisons couramment.

La rapidité de modulation s'exprime en **bauds**. Elle correspond au nombre de changements d'états du signal par seconde sur la ligne de transmission. Une rapidité de $b$ bauds ne correspond pas forcément à $b$ bits/s sur la ligne. Une configuration binaire (un ou plusieurs bits selon la valence) correspond à un état du signal.

### 4.5 Le dialogue

Intéressons-nous maintenant au dialogue entre l'équipement informatique et la jonction.

#### 4.5.1 La jonction série

La jonction spécifie les caractéristiques mécaniques, électriques et fonctionnelles des signaux. Bien entendu ces jonctions sont normalisées (voir plus loin les tableaux récapitulatifs sur l'état actuel de la normalisation) et celle qui nous intéresse plus particulièrement est référencée sous le nom V.24 par l'U.I.T, sensiblement équivalente de la norme RS-232C définie par l'E.I.A[3].

Voici une description des signaux de l'interface V.24 les plus couramment utilisés :

| Code | No broche ISO 2110 | No broche DB 9 | RS-232 | V.24 | Signification |
|---|---|---|---|---|---|
| 101 | 1 | | PG | TP | Terre de protection |
| 102 | 7 | 5 | SG | TS | Terre de signalisation |
| 103 | 2 | 3 | TD | ED | Emission de donnees |
| 104 | 3 | 2 | RD | RD | Reception de donnees |
| 105 | 4 | 7 | RTS | DPE | Demande pour emettre |
| 106 | 5 | 8 | CTS | PAE | Pret a emettre |
| 107 | 6 | 6 | DSR | PDP | Poste de donnees pret |
| 108 | 20 | 4 | DTR | TDP | Terminal de donnees pret |

[3]Electronic Industries Association.

```
| 109  |     8     |     1     |  DCD   | DS   |  Detection du signal de ligne  |
+------+-----------+-----------+--------+------+--------------------------------+
| 125  |    22     |     9     |  RI    | IA   |  Indicateur d'appel            |
+------+-----------+-----------+--------+------+--------------------------------+
```

Brochage des prises côté soudures :

```
+-----------+          +---------------------------------------+
| 5 4 3 2 1 |          | 13 12 11 10 9  8  7  6  5  4  3  2  1 |
\  9 8 7 6  /          \  25 24 23 22 21 20 19 18 17 16 15 14  /
 `---------'            `-------------------------------------'
    DB 9                              ISO IS 2110
```

### 4.5.2 Le dialogue proprement dit

Prenons deux postes de travail équipés d'un modem chacun et souhaitant communiquer.

Nous passerons rapidement sur le fait que les équipements doivent être reliés à la masse. Ceci est réalisé grâce au circuit 101. D'autre part, il est nécessaire de définir une référence de signalisation : c'est le rôle du circuit 102.

Dès sa mise sous tension, l'ETTD présente un état logique "1" sur le circuit 108 : *Terminal de Données Prêt* (DTR). Dès la mise sous tension de l'ETCD, celui-ci présente l'état *Poste de Données Prêt* (DSR) correspondant à un état logique "1" sur le circuit 107, assurant ainsi que le modem est sous tension et connecté à la ligne.

L'ETTD ayant des données à émettre, demande à émettre. Il présente sur la jonction l'information *Demande Pour Émettre* (RTS) sur le circuit 105. Ceci valide le modulateur de l'ETCD qui émet alors une porteuse.

Du coté appelé, l'ETCD détecte la présence de la porteuse sur la ligne de transmission et le signale à l'ETTD sur le circuit 109 : *Détection de signal* (porteuse). Les circuits 107 et 108 auront été initialisés au préalable comme ci-dessus.

L'ETTD ayant signalé son intention d'émettre sur le circuit 105 reçoit en réponse peu de temps après le signal *Prêt À Émettre* (CTS) sur le circuit 106.

Les données peuvent ensuite circuler via les circuits 103 et 104.

### 4.5.3 Le contrôle de flux

Lorsqu'un émetteur émet de façon systématique plus de données que le récepteur ne peut en accepter, il se pose alors un problème qui ne peut être résolu que grâce au mécanisme de *contrôle de flux*.

Le contrôle de flux peut être de différents types :

**logiciel**

Le modem insère des caractères de contrôles dans le flot de données circulant entre l'ETCD et l'ETTD : **XOFF** pour arrêter l'envoi et **XON** pour le reprendre.

**matériel**

Généralement appelé **CRTSCTS**, il met en oeuvre l'emploi des circuits 105 (RTS) et 106 (CTS). Ce symbole est en fait le nom donné à la constante du fichier d'inclusion *termios.h*.

Le fonctionnement du contrôle de flux matériel pendant la transmission peut se résumer ainsi :

Avant d'émettre, le terminal doit lever son signal RTS (Request To Send). À partir de ce moment, le modem, s'il est en mesure d'émettre, lève le signal CTS (Clear To Send). Lorsque le buffer du modem est plein, le modem descend CTS. Il le remonte ensuite. Dans l'autre sens de transmission, lorsque le buffer du terminal est plein, le terminal descend RTS.

## 4.6 La connexion au réseau téléphonique commuté

Maintenant, plusieurs questions se posent, et j'imagine que parmi celles que vous vous posez il y a :

- et sous Linux, le fonctionnement est-il identique ?
- à quel moment le numéro du correspondant a t-il été composé ?
- mon modem est configuré en réception/émission, comment ça marche ?
- etc.

Nous allons maintenant tenter de répondre.

Eclaircissons un peu les choses. Le dialogue que nous venons de voir concerne le dialogue *théorique* ETTD-ETCD et ETCD-ETTD sans se soucier d'éventuelles contraintes pouvant provenir du système d'exploitation. Il est toujours vrai. Néanmoins, il ne suffit pas forcément pour qu'une communication soit établie, notamment via le RTC. Nous allons étudier ce fonctionnement point par point en prenant un bon système d'exploitation (**Linux**, mais ce n'est qu'un exemple), un bon port série et du courage. Vous continuez ?

Tout d'abord, nous avons vu qu'une communication commençait toujours par le premier échange DTR/DSR, ou si vous préférez 108/107. La montée du circuit 108 est réalisée sous Linux à l'ouverture du port série (ex. `fopen ("/dev/ttyS0", ...)`). Cela se voit très bien sur un modem externe, le voyant TR est allumé. La réponse du modem par le circuit 107 est un peu différente. Dans la section 4.5.2 (Le dialogue proprement dit), pour des raisons de simplicité, nous supposions que le modem répondait sur le circuit 107 après un délai très bref, c'est-à-dire qu'il était instantanément connecté à la ligne.

Cette réponse est maintenant conditionnée par la connexion à la ligne via le réseau téléphonique commuté.

### 4.6.1 Initialisation du modem

En général, c'est juste après l'ouverture du port série que le modem est initialisé. Cela se fait grâce aux commandes AT que nous ne détaillerons pas. Simplement, ces commandes sont envoyées au modem (par l'intermédiaire du circuit 103) (ex. `write` sur le *descripteur de fichier* du périphérique) et interprétées par lui, lorsque :

- le circuit 108 est fermé (état "1") ;
- le modem est en mode commande.

### 4.6.2 Établissement de la connexion

L'une des commandes d'initialisation permet la composition d'un numéro. Le modem décroche (eh oui, ce terme barbare veut dire que suite à la fermeture du relais, le central local envoie une tonalité à la fréquence de 440 Hz `:-)`) puis compose le numéro.

Sur l'équipement distant, le circuit 108 est également monté. Le modem appelé détecte l'appel. Le signal d'indication d'appel (circuit 125) est utilisé en interne pour mémoriser l'appel, le modem réalisant donc lui-même la connexion à la ligne. Cette mémorisation est maintenue par DTR (jusqu'à déconnexion).

À ce moment précis, le modem appelé répond en validant son modulateur qui émet la porteuse.

Le modem appelant, en état de décrochage et attendant la porteuse, met son émetteur en service. Après négociation, le circuit 109 (DCD) est alors validé. Du côté de l'appelé, le circuit 109 est également validé. La prise de contact est terminée. Les circuits 107 (DSR) des deux modems sont alors montés en réponse à DTR (asservissement des circuits 107-109).

### 4.6.3 Réponse automatique ou manuelle

Du côté de l'appelé, il est possible de mettre le modem en mode réponse automatique. Il répond alors tout seul à l'appel après quelques sonneries. Le registre S0 des modems est généralement réservé à la configuration de ce mode.

Lorsque ce registre contient la valeur 0, (ATS0=0), le modem est en réponse manuelle. Sous Linux, c'est assez souvent l'option choisie, et c'est le logiciel (notamment *getty*) qui gère l'appel. En effet les gestionnaires, de *tty* tels que *getty* préfèrent prendre en charge la connexion : ce n'est pas au modem à répondre à un appel mais à *getty* lui même. Lorsque le modem reçoit un appel, il émet simplement le message **RING** (bien sûr, si le mode verbeux est bien configuré : ATE1). Sur ce, *getty* envoie la commande ATA qui valide le mode réponse et la porteuse.

Enfin, à la fermeture du port, les signaux 108/107 repassent à l'état **0**.

Voici à titre d'information l'organigramme d'un appel :

```
                    +---------------------------------------+
                    | Detection de l'invitation a numeroter |
                    +---------------------------------------+
                                       |
                          +---------------------------+
                          |       Numerotation        |
                          +---------------------------+
                                       |
                       +---------------------------------+
                       | Emission de la tonalite d'appel |
                       +---------------------------------+
                                       |
                          +---------------------------+
+------- NON -----|   Detection de tonalite   |- OUI
|                 +---------------------------+    |
|                     |                             |
|         +--------------+           +---------------------------+
|         |   Occupe     |           |     Retour d'appel        |
|         +--------------+           +---------------------------+
|           |                                      |
|           |                        +---------------------------+
+------------------------- NON -------| Detection arret tonalite |
|           |                        +---------------------------+
|           |                                      |
|           |                     +--------------------------------+
|           |    +-------- NON ----| Detection tonalite de reponse |
|           |    |                 +--------------------------------+
|           |    |                                 |
|           |    |                   +---------------------------+
|           |    |            +-NON--| Prise de contact aboutie |
```

```
        |           |    |            |      +-------------------------+
        |           |    | NO         |                        |
 TIMEOUT |    BUSY  |    | ANSWER     | TIMEOUT                | CONNECT
    +-------------------------+   +-------------+  +-------------------------+
    |    Appel infructueux     |   | Deconnexion |  | Transmission de donnees  |
    +-------------------------+   +-------------+  +-------------------------+
```

### 4.6.4 Déconnexion

Plusieurs méthodes permettent de mettre fin à un appel :

- **Mode commande.** Mettre le modem en mode commande et envoyer la chaîne `+++ATH` ;
- **Perte de porteuse.** L'une des causes est la déconnexion normale de l'autre modem ;
- **Ouverture du circuit 108** (DTR). C'est la méthode la plus couramment employée.

# 5 Etat actuel de la normalisation

Voici un tableau résumant l'état actuel de la normalisation concernant les classes de débits supportés.

```
+----------+------------------------------------------+
|   Avis   |              Signification               |
+----------+------------------------------------------+
| V.21     | Utilisation sur RTC a 300 bits/s         |
|          |                                          |
| V.22     | Utilisation a 1200 bits/s sur RTC 2 fils |
|          | full duplex                              |
| V.22 bis | idem a 2400 bits/s                       |
|          |                                          |
| V.23     | Utilisation a 600 ou 1200 bits/s sur RTC |
|          | ou 1200/75 bits/s                        |
|          |                                          |
| V.25 et  | Composition automatique du numero et/ou  |
| V.25 bis | reponse automatique a un appel sur RTC   |
|          |                                          |
| V.26     | Utilisation a 2400 bits/s sur LS(*) 4 fils |
|          |                                          |
| V.26 bis | Modem 2400 bits/s (1200 en repli) sur RTC |
|          |                                          |
| V.27     | Modem 4800 bits/s pour LS                |
|          |                                          |
| V.27 bis | Modem 4800 bits/s (2400 en repli) pour   |
|          | donnees synchrones                       |
|          |                                          |
| V.27 ter | Modem 4800 bits/s (2400 en repli) meme   |
|          | modulation mais pour le RTC              |
|          |                                          |
| V.29     | Modem 9600 bits/s pour LS                |
|          |                                          |
| V.32     | 9600 bits/s (4800 en repli) duplex 2 fils |
|          | sur RTC                                  |
```

| | |
|---|---|
| V.32 bis | 14400 bits/s |
| V.34 | 28800 bits/s sur RTC |
| V.42 | Correction d'erreurs LAP-M et MNP4 |
| V.42 bis | Correction d'erreurs +<br>compression de donnees MNP5 |
| V.54 | Normalise les boucles de tests |

* LS = Ligne Specialisee

## 5.1 À propos du V.42 bis

Un tout petit mot à propos de la norme V.42 bis qui permet la compression de données. L'algorithme utilise un dictionnaire de chaînes de caractères. Lorsqu'une chaîne apparaît, un *token* est transmis qui n'est autre que l'index de cette chaîne dans le dictionnaire. La longueur maximale d'une chaîne ainsi que la taille maximale du dictionnaire sont négociées au début de la connexion. La norme V.42 bis autorise une longueur de chaîne comprise entre 6 et 250 caractères. La taille minimale du dictionnaire est de 512 entrées (soit 9 bits pour coder le rang d'un entrée). Le taux maximal de compression dans ce cas est de :

```
250 * 8 : 9 = 222.2
```

soit un taux de 222:1. Un bon taux de compression est plus une affaire de taille mémoire et d'efficacité en fonction des données à coder qu'une affaire de puissance de processeur.

## 5.2 Débits et modulations

Merci à Christian 'naddy' Weisgerber de son aide pour la rédaction de cette partie.

Voici rassemblés dans les tableaux suivants les débits et les modulations correspondantes utilisés dans les principales normes pour liaisons téléphoniques à 2 fils. Les *normes* qui ne sont pas citées ci-après sont peu utilisées voire oubliées aujourd'hui (liaisons à 4 fils, V.32 terbo, Bell xxx, V.FC, ZyXEL, HST, PEP... certaines n'étant d'ailleurs pas de véritables normes).

| Avis | b/s | bauds | modulation | remarques |
|---|---|---|---|---|
| V.21 | 300 | 300 | FSK | |
| V.22 | 1200 | 600 | DPSK | |
| V.22bis | 2400 | 600 | QAM | |
| V.23 | 1200 | 1200 | FSK | |
| | 600 | 600 | FSK | |
| | 75 | 75 | FSK | [1] |
| V.32 | 9600 | 2400 | QAM+TCM | |

| | | | | |
|---|---|---|---|---|
| | 9600 | 2400 | QAM | |
| | 4800 | 2400 | QAM | |
| | | | | |
| V.32bis | 14400 | 2400 | QAM+TCM | |
| | 12000 | 2400 | QAM+TCM | |
| | 9600 | 2400 | QAM+TCM | |
| | 7200 | 2400 | QAM+TCM | |
| | 4800 | 2400 | QAM | |
| | | | | |
| V.34 | (voir tableau suivant) | | | |
| | | | | |
| V.27ter | 4800 | 1600 | DPSK | |
| | 2400 | 1200 | DPSK | |
| | | | | |
| V.29 | 9600 | 2400 | QAM | |
| | 7200 | 2400 | QAM | |
| | 4800 | 2400 | QAM | [2] |
| | | | | |
| V.17 | 14400 | 2400 | QAM+TCM | |
| | 12000 | 2400 | QAM+TCM | |
| | 9600 | 2400 | QAM+TCM | |
| | 7200 | 2400 | QAM+TCM | |

[1] Bande de retour.
[2] Pas utilise pour fax.

V.21, V.22, V.22bis, V.32, V.32bis, V.34 sont "full duplex".
V.27ter, V.29, V.17 sont "half duplex" et utilises pour fax.
V.23 est "half duplex" et asymetrique.

Les modulations:

FSK : Frequency Shift Keying (modulation de frequence)
DPSK : Differential Phase Shift Keying (modulation de phase differentielle)
QAM : Quadrature Amplitude Modulation (modulation d'amplitude en quadrature)
TCM : Trellis Coded Modulation (modulation codee en treillis)

Dans le cas de l'avis V.34, les choses se compliquent un peu. Cette norme a des vitesses de modulation obligatoires (2400, 3000, 3200 bauds) et des vitesses facultatives (2743, 2800, 3429 bauds). La modulation est toujours de type QAM (modulation d'amplitude en quadrature) avec une des trois méthodes TCM choisie par le récepteur. Les combinaisons suivantes sont possibles :

| b/s | 2400 | 2743 | 2800 | 3000 | 3200 | 3429 bauds |
|---|---|---|---|---|---|---|
| 2400 | x | | | | | |
| 4800 | x | x | x | x | x | x |
| 7200 | x | x | x | x | x | x |
| 9600 | x | x | x | x | x | x |
| 12000 | x | x | x | x | x | x |
| 14400 | x | x | x | x | x | x |
| 16800 | x | x | x | x | x | x |
| 19200 | x | x | x | x | x | x |

```
| 21600     x       x       x       x       x       x          |
| 24000             x       x       x       x       x          |
| 26400                             x       x       x          |
| 28800                                     x       x          |
+--------------------------------------------------------------+
```

# 6 Foire Aux Questions

**Comment puis-je changer facilement un paramètre de mon port série ?**

La meilleure façon de le faire, aussi bien manuellement que dans un script est de rediriger le périphérique sur l'entrée standard de *stty*. Exemple :

```
stty crtscts < /dev/ttyS0
```

activera le contrôle de flux matériel sur le premier port série utilisé en entrée.

```
stty -a < /dev/cua0
```

affichera tous les paramètres du premier port série utilisé en sortie.

**Pourquoi faut-il configurer CRTSCTS sur le port série ?**

Pour gérer le contrôle de flux matériel. Ce n'est pas une obligation, c'est une garantie que l'échange de données entre ETTD et ETCD se fera dans les meilleures conditions. Il faut bien entendu que votre modem puisse le faire. Contrairement à une idée reçue, si vous mettez l'option CRTSCTS dans le fichier /etc/gettydefs, il n'est pas nécessaire d'effectuer en plus un `stty crtscts </dev/port\#`. Par contre, il faut le mettre à la fois dans la partie *initiale* et *finale* de gettydefs. Notez qu'il s'agit bien d'un contrôle de flux local, et en aucun cas il ne faut s'inquiéter de ce que fait le correspondant dans ce domaine.

**Je remarque que agetty modifie les droits du fichier /dev/ttyS0, bizarre non ?**

Ca peut effectivement paraître bizarre. Il s'agit en fait de l'établissement d'une *session* d'utilisation du périphérique. Celui-ci prend alors les droits du "chef" de session (*session leader*) qui se protège ainsi des utilisations du même tty par d'autres processus.

**Aurais-je accès à mon téléphone ?**

Cette question a été réellement posée. Si vous n'avez qu'une ligne téléphonique, la réponse est non. De plus, en décrochant le combiné téléphonique, vous pertuberez la ligne et le modem risque fort de diminuer le débit (pour le remonter si tout va bien ensuite).

**Lorsque je me connecte chez mon fournisseur, comment mon adresse IP est générée ?**

Cette question montre à l'évidence une confusion entre toutes les notions réseaux. Bien qu'elle ait été posée suite à un problème relatifs aux modems, la réponse ne devrait théoriquement pas se trouver dans ce document. Néanmoins, le chapitre suivant rappelle quelques principes de base des empilements protocolaires afin de clarifier un peu tout cela.

# 7 Un mot sur les empilements protocolaires couramment utilisés

Un tel titre pourrait faire croire à une erreur de mise en page ou de *copier-coller* étant donné le sujet du document. En fait, il n'en est rien.

La connexion d'une machine à un fournisseur d'accès à Internet met en jeu un ensemble de protocoles de communications : TCP, UDP, IP, SLIP, PPP, etc. De nombreux utilisateurs souhaitent réaliser ce type de connexion depuis chez eux, via un modem et rencontrent parfois quelques problèmes de configuration.

Il est évident qu'il est à la fois difficile et inutile de tout connaître de ces protocoles. Il faut vraiment *être du métier* pour bien les connaître, et encore ! Cependant il semble raisonnable de penser que la mise en oeuvre de telles connexions, sous Linux par exemple, ne peut se faire dans de bonnes conditions sans un minimum de connaissances sur l'architecture de communication utilisée.

La lecture du forum *fr.comp.os.linux* montre parfois une certaine confusion dans toutes les fonctions mises en oeuvre et qui engendrent inévitablement de mauvais paramétrages.

Les quelques schémas qui suivent donnent une idée de la façon dont tous ces *engrenages* sont placés pour que "ça tourne" !

## 7.1 TCP/UDP/IP

Ces sigles sont très fréquemment utilisés aujourd'hui et pour cause : ces empilements de couches de communications tendent à se répandre à vive allure. C'est à l'origine un ensemble de protocoles développés dans le cadre du projet ARPANET, créé par ARPA (aujourd'hui DARPA), l'agence pour les projets de recherche avancée du Ministère de la Défense des Etats-Unis.

- TCP[4] est une entité de niveau Transport chargée de véhiculer des données de manière fiable entre deux machines souhaitant dialoguer ;
- UDP[5] est une entité de niveau Transport chargée de véhiculer des données entre deux machines souhaitant dialoguer ;
- IP[6] est une entité de niveau Réseau chargée de véhiculer des données entre deux noeuds d'un réseau.

## 7.2 PPP/SLIP

PPP[7] et SLIP[8] proposent une méthode d'encapsulation des datagrammes IP sur des liaisons point à point, par exemple les lignes asynchrones série. En quelques mots, disons que SLIP est un protocole très simple, assez ancien, datant d'une époque où certains problèmes n'étaient pas aussi importants qu'aujourd'hui : adressage, identification réciproque, détection et correction d'erreurs, compression (extrait du RFC-1055). PPP est quant à lui beaucoup plus complet et c'est pourquoi il est généralement préféré par les connaisseurs. Il offre toutes ces caractéristiques regroupées en trois sous-ensembles :

- une méthode d'encapsulation de type HDLC sur circuit commuté ou permanent, synchrone ou asynchrone ;
- un protocole LCP (Link Control Protocol) permettant d'établir, de configurer et de tester une connexion ;
- une famille de protocoles NCP (Network Control Protocols) pour l'établissement et la configuration des protocoles réseaux.

Pour obtenir de plus amples renseignements, vous pouvez vous reporter aux documents concernant ces protocoles : **RFC-1055** (SLIP), **RFC-1171** et **RFC-1172** (PPP) et le **PPP-HOWTO**.

[4] Transmission Control Protocol
[5] User Datagram Protocol
[6] Internet Protocol
[7] Point-to-Point Protocol
[8] Serial Line IP

## 7.3 Mise en oeuvre

```
+----------------+-------+ +--------+ +--------+ +------+ +-----+
|  Applications  |  FTP  | | TELNET | |  SMTP  | | TFTP | | ... |
+----------------+---+---+ +---+----+ +--------+ +--+---+ +-----+
                     |         |      |             |
+----------------+---+---------+------++------------+-----+
| Transport      |         TCP         |       UDP        |
+----------------+---------------------+------------------+
| Reseau         |                                        |
|                |                   IP                   |
|                |     -----         -----     ------     |
| (interfaces)   |    / eth \       / ppp \   / slip \    |
+----------------+-----------------^---------+------------+
| Liaison        |       LLC       |   PPP   |   SLIP     |
|                |       MAC       |         |            |
+----------------+-------++--------+--------+-+-----------+
                         ||                 | |
+----------------+-------++--------+--------+-+-----------+
| Interfaces     |      LAN        |       V.24           | Fonctionnelle
|                |  (non decrit)   |       V.28           | Electrique
|                |                 |  ISO.2110 (25 br)    | Physique
+----------------+-------++--------+--------+-+-----------------+
                         ||                 | |
              Ethernet  +--+              +-+-+-+  Modem
              ==========|  |=====         |\     \-------
              Fddi      +--+              \ \_____\     /
                                           \|_0_0_|    ---------
```

La mise en oeuvre de tels protocoles se fait toujours de bas en haut (des couches basses aux couches hautes) puisque la demande se fait de haut en bas. Prenons un exemple :

Supposons que vous souhaitiez faire un *ftp* de chez vous sur *ftp.samachine.fr*. L'application *ftp* demande à TCP d'établir une connexion. Pour qu'elle puisse s'établir, PPP doit déjà fonctionner. Pour que PPP fonctionne, le modem doit être en ligne :

- première étape : établissement d'une communication entre deux modems. Cette étape est supposée connue, maintenant ;
- deuxième étape : mise en route de PPP, avec éventuellement authentification ;
- troisième étape : configuration de l'interface IP correspondante. Il s'agit en général, sous Linux, de l'interface *ppp0*. Une adresse IP pouvant provenir soit d'une configuration locale, soit de votre fournisseur, est affectée à l'interface. Dans ce dernier cas, celui-ci la fournit lors de l'initialisation PPP. Dans les deux cas, c'est le démon PPP qui configure l'adresse de l'interface ;
- quatrième étape : établissement d'une connexion TCP, puis initialisation de *ftp*. A partir de là, si tout s'est bien passé, vous pouvez transférer vos fichiers.

Vous pouvez maintenant imaginer le déroulement d'une déconnexion.

## 7.4 Les fichiers de configuration

### 7.4.1 Les applications

Vous comprendrez qu'il est difficile de décrire ici l'emplacement des fichiers de configuration des applications. Prenez soin de lire les fichiers README ou INSTALL et d'exécuter l'installation correctement.

### 7.4.2 Les couches de communication : TCP/UDP/IP

Pour une configuration standard de votre machine, vous devez compiler le noyau avec les options "réseau" suivantes :

```
Networking support              y

Network firewalls               n
Network aliasing                n
TCP/IP networking               y

IP: forwarding/gatewaying       n
IP: multicasting                n
IP: accounting                  n
IP: PC/TCP compatibility mode   n
IP: Reverse ARP                 n
IP: Disable Path MTU Discovery
        (normally enabled)      n
IP: Disable NAGLE algorithm
        (normally enabled)      n
IP: Drop source routed frames   y
IP: Allow large windows
        (not recommended
        if <16Mb of memory)     n

The IPX protocol                n
... autres protocoles           n
```

Le fichier *resolv.conf* doit contenir :

```
domain <domaine de votre fournisseur>
nameserver <adresse IP du serveur de nom de votre fournisseur>
```

# 8 Le Minitel

Bien que ce merveilleux appareil commence à prendre de l'âge, il est difficile de ne pas en parler un peu, notamment en raison de ses spécificités. Pourquoi ne pas envisager en effet de faire un serveur Minitel chez vous ou tout simplement de l'utiliser comme terminal. Nous nous contenterons ici d'en donner quelques caractéristiques intéressantes dans le cadre d'une utilisation avec Linux.

Les STUM 1B[9] décrivent l'ensemble des caractéristiques des divers modules du Minitel 1B :

- l'écran ;

[9] Spécifications Techniques d'Utilisation du Minitel 1B.

- le clavier ;
- le modem ;
- la prise péri-informatique.[10]

## 8.1 L'écran

Le minitel 1B est capable d'afficher 24 lignes de 40 ou 80 caractères et 8 couleurs (ou niveaux de gris).

Le mode 40 colonnes correspond au standard *Videotex*, le mode 80 colonnes au standard *télé-informatique*. C'est en général celui-ci que l'on utilisera s'il sert de terminal. Les séquences de touches permettant de passer d'un mode à l'autre sont indiquées dans le tableau suivant dans lequel on retrouvera également quelques séquences utiles :

```
+-------------+----------------------------------+
|   Touches   |           Signification          |
+-------------+----------------------------------+
| <Fcnt T> A  | Mode tele-informatique americain |
|             | (pas d'accents)                  |
| <Fcnt T> F  | Mode tele-informatique francais  |
|             | accents (codage particulier)     |
| <Fcnt T> V  | Mode Videotex                    |
+-------------+----------------------------------+
| <Fcnt T> E  | Valide/invalide l'echo local     |
|             |                                  |
| <Fcnt E> P  | Mode page (retour haut de page   |
|             | en fin d'ecran)                  |
| <Fcnt E> R  | Mode rouleau (par defaut)        |
|             |                                  |
| <Fcnt C> M  | Verouillage minuscules (defaut   |
|             | en mode tele-informatique)       |
+-------------+----------------------------------+
```

## 8.2 Le clavier

Il s'agit d'un clavier *AZERTY* permettant la saisie de la plupart des caractères courants pour un terminal. Il est notamment possible de verouiller les minuscules grâce à la séquence <Fcnt C> M. A noter une correspondance, dans le mode télé-informatique, de certaines touches :

```
+-------------+----------------------------------+
|   Touches   | Correspondance terminal classique |
+-------------+----------------------------------+
| Sommaire    |   PF1                             |
|             |                                   |
| Annulation  |   PF2                             |
|             |                                   |
| Retour      |   PF3                             |
|             |                                   |
| Repetition  |   PF4                             |
|             |                                   |
```

[10] A ce propos, je tiens à votre disposition un schéma électronique d'un montage permettant l'adaptation RS232-Minitel. Il met en oeuvre le circuit MAX232 permettant une parfaite adaptation des tensions.

```
| Envoi       |   Enter (Entree)                  |
+-------------+-----------------------------------+
```

La touche *Entrée* correspond également à la séquence de touches <Ctrl J>

## 8.3 Le modem

Le modem du minitel permet des débits de 300 à 4800 ou 9600 bits/s[11]. Il est associé à un coupleur travaillant sur 7 bits de données, un bit de parité paire, un bit de *start* et un bit de *stop*, soit 10 bits par caractère. Le tableau suivant donne les séquences de touches permettant de configurer le modem à ces différents débits.

```
+-------------+--------------+
|   Touches   |    Debit     |
+-------------+--------------+
| <Fcnt P> 3  |  300 bits/s  |
|             |              |
| <Fcnt P> 1  | 1200 bits/s  |
|             |              |
| <Fcnt P> 4  | 4800 bits/s  |
|             |              |
| <Fcnt P> 9  | 9600 bits/s  |
+-------------+--------------+
```

En standard V.23, il est possible de *retourner* le modem (vitesse émission-réception) avec la séquence <Fcnt M> R.

## 8.4 Utilisation du Minitel comme simple terminal

D'après les conseils avisés de *Pierre Ficheux*, voici un exemple de configuration permettant de connecter un Minitel :

### 8.4.1 Configuration de getty

Une méthode simple consiste à compiler un *getty* un peu particulier. Les sources se trouvent dans le paquetage **getty_ps-2.0.7h**, en général disponible par *ftp* (ftp.ibp.fr) sous **/pub/linux/tsx-11/sources/sbin**.

Il s'agit ensuite de modifier le fichier *tune.h* comme suit :

```
#ifdef V23
#define DEF_CFL  (CS7|PARENB)                   /* Pour connexion V.23 */
#else
#define DEF_CFL  (CS8)                  /* default word-len/parity */
#endif /* V23 */
```

Puis de compiler l'ensemble avec l'option -DV23 vous donnant un fichier exécutable *uugetty* que vous pourrez renommer *uugetty_v23*. Ensuite, il faut ajouter quelques entrées au fichier */etc/gettydefs* :

```
#
# Pour la connexion V.23
#
```

[11] Tous les modèles de minitel n'autorisent pas tous ces débits.

```
9600v23# B9600 CS7 PARENB -PARODD CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S login: #4800v23

4800v23# B4800 CS7 PARENB -PARODD CLOCAL # B4800 SANE -ISTRIP CLOCAL #@S login: #2400v23

2400v23# B2400 CS7 PARENB -PARODD CLOCAL # B2400 SANE -ISTRIP CLOCAL #@S login: #1200v23

1200v23# B1200 CS7 PARENB -PARODD CLOCAL # B1200 SANE -ISTRIP CLOCAL #@S login: #1200v23
```

Enfin, vous modifiez le fichier *inittab* de façon à démarrer *uugetty_v23* comme dans l'exemple ci-dessous :

```
d4:45:respawn:/sbin/uugetty_v23 ttyS1 9600v23
```

Une solution différente consisterait à seulement modifier */etc/gettydefs*. La méthode précédente ne fait, finalement, que redéfinir l'option SANE qui ne comporte pas moins de 16 paramètres, en modifiant l'un d'eux : DEF_CFL. Il est possible de la redéfinir par configuration en la remplaçant par l'ensemble des paramètres, sauf CS8 que l'on remplacera par CS7 PARENB. Les entrées de */etc/gettydefs* sont à modifier comme l'exemple ci-après :

```
#
# Pour la connexion V.23
#
9600v23# B9600 CS7 PARENB -PARODD CLOCAL # B9600 ISTRIP CS7 PARENB -PARODD (*)
 CLOCAL BRKINT IGNPAR ICRNL IXON IXANY OPOST ONLCR CREAD HUPCL ISIG ICANON (*)
 ECHO ECHOE ECHOK #@S login: #4800v23

(*) a continuer sur la meme ligne

[reste du ficher]
```

Bien que cela puisse paraître lourd, il est préférable d'utiliser cette méthode. On a trop tendance à recompiler des sources pour les adapter à trente-six situations alors qu'ils fournissent généralement un niveau de configurabilité extrêmement complet et puissant. C'est le cas ici. Les sources de *getty* prévoient d'analyser tous ces paramètres, et cela fonctionne parfaitement.

### 8.5 Utilisation du Minitel pour un accès distant

Deux cas peuvent se présenter :

- soit vous souhaitez dédier votre ligne aux seuls accès Minitel, auquel cas il est souhaitable d'envisager de configurer votre système comme ci-dessus. Il faut simplement modifier le fichier */etc/gettydefs* en remplaçant l'option CLOCAL par CRTSCTS dans la première partie (options initiales) et par CRTSCTS HUPCL dans le partie suivante (options finales) ;
- soit vous souhaitez seulement *pouvoir* utiliser le Minitel comme terminal distant, sans que ce soit un accès dédié. Dans ce cas il n'y a rien à faire sur votre système (rien de plus que la configuration que vous avez déjà dû mettre en place : getty, gettydefs, inittab, ...). Sur le Minitel, il sera préférable de le configurer en mode télé-informatique (<Fcnt T> F).

## 9 Un modem particulier: le câble null-modem

Un câble *null-modem* est tout simplement un câble inverseur permettant de relier ensemble deux ETTD sans passer par l'intermédiaire de deux ETCD. Il est conforme aux normes du C.C.I.T.T., c'est à dire qu'il simule les différents signaux. Son seul inconvénient est que la liaison ainsi réalisée ne peut dépasser 250 mètres.

Voici le schéma du câble à réaliser :

```
+------+-------+---------------------------------+------+-------+
| Code | V.24  |     Cablage (No de broche)      | Code | V.24  |
+------+-------+---------------------------------+------+-------+
| 101  |  TP   |    1 o--------------------o 1   | 101  |  TP   |
|      |       |                                 |      |       |
| 102  |  TS   |    7 o--------------------o 7   | 102  |  TS   |
|      |       |                                 |      |       |
| 103  |  ED   |    2 o--------------------o 3   | 104  |  RD   |
|      |       |                                 |      |       |
| 104  |  RD   |    3 o--------------------o 2   | 103  |  ED   |
|      |       |                                 |      |       |
| 105  |  DPE  |    4 o-|               |-o 4   | 105  |  DPE  |
|      |       |        |               |        |      |       |
| 106  |  PAE  |    5 o-|               |-o 5   | 106  |  PAE  |
|      |       |        \        ,------'        |      |       |
|      |       |         -------(---------o 8   | 109  |  DS   |
| 109  |  DS   |    8 o---------'                |      |       |
|      |       |                                 |      |       |
| 107  |  PDP  |    6 o-------------------o 20  | 108  |  TDP  |
|      |       |                                 |      |       |
| 108  |  TDP  |   20 o-------------------o 6   | 107  |  PDP  |
|      |       |                                 |      |       |
+------+-------+---------------------------------+------+-------+
```

# 10 Choix d'un modem

Si vous êtes sur le point d'investir dans un modem, il y a au moins trois facteurs déterminants pour vous :

**Le coût**

Ce facteur est généralement le plus important. Vous vous êtes certainement fixé une somme que vous ne pourrez dépasser que dans des limites raisonnables (moins de 500 francs environ) et moyennant un service supplémentaire non négligeable. On trouve à l'heure actuelle d'excellents modems dans une gamme de prix de l'ordre de 1000 FF à 2000 FF.

**Le débit**

La réponse pourrait être vague : tout dépend de l'utilisation. En fait, il semble que la plupart d'entre vous se connectera à Internet et la plupart des fournisseurs proposent désormais des accès à 28800 bits/s. Il semble raisonnable d'investir dans un tel modem, d'autant plus que leur coût est rarement supérieur à 2000 FF. Vous pouvez d'ailleurs espérer gagner un peu sur les temps de connexions lors de transferts de fichiers ou de navigation "WEB".

**Le type de modem : interne ou externe**

Pour répondre rapidement, on pourrait dire que le modem interne a, en France, tous les défauts. Hors de France, il a beaucoup de défauts. C'est un avis, il vaut ce qu'il vaut, mais il semble partagé par bon nombre d'utilisateurs :

- le premier défaut est qu'il est interne et occupe donc un emplacement sur le *bus*. En externe le modem est de plus facilement transportable d'un PC à un autre, par exemple ;

- le deuxième est que notre opérateur national demande à ce qu'un numéro soit “brûlé” lorsqu'un appel échoue plusieurs fois. Une fois le numéro brûlé, le modem refuse de le composer. La seule solution est alors d'éteindre le modem et de recommencer. Au fait, comment éteignez-vous un modem interne ?
- le troisième est plutôt lié au confort. La plupart des modems externes possèdent en effet des voyants indiquant l'état de la connexion. Il est agréable de constater rapidement que la connexion est perdue ou que le transfert est perturbé.

Le seul inconvénient du modem externe est qu'il occupe un port série. De plus il vaut mieux posséder un port série rapide de type 16550A (les 16550 sont bogués). Notez cependant que les ports série de type 8250 (les anciens) supportent des débits pouvant aller jusqu'à 57600 bits/s. Contrairement à ce que l'on dit, ils ne sont pas si mauvais que cela. Qui est déjà monté à des débits supérieurs ?

# 11 Quelques chiffres

Beaucoup d'utilisateurs se posent des questions sur ce qu'ils peuvent attendre de leur modem et du coût des communications. Bien souvent, la facture surprend tous les deux mois.

## 11.1 Débit réel

Un petit tableau vaut mieux qu'un long discours, voici quelques valeurs de débits réels. Evidemment, il n'est pas tenu compte des problèmes de lignes, de disponibilité des serveurs qui réduisent parfois les taux de transfert effectifs. De plus, les valeurs sont indiquées sans compression logicielle. En cas de compression, il faudra multiplier les valeurs par au plus 2, ce qui est déjà optimiste.

Il faut savoir qu'en transmission asynchrone 8 bits, étant donnés les bits de **start** et de **stop**, le rendement est d'environ 80%.

| Debit affiche | Debit reel | Ko/s | Mo/heure |
|---|---|---|---|
| 9600 b/s | 7680 b/s | 0,96 | 3,4 |
| 14400 b/s | 11520 b/s | 1,44 | 5,2 |
| 19200 b/s | 15360 b/s | 1,92 | 6,9 |
| 21600 b/s | 17280 b/s | 2,16 | 7,8 |
| 26400 b/s | 21120 b/s | 2,64 | 9,5 |
| 28800 b/s | 23040 b/s | 2,88 | 10,4 |

## 11.2 Coûts de connexion

Le meilleur indicateur pour ce genre de calcul est bien entendu France Télécom. Essayons ici de dégager quelques grands chiffres. Les prix sont indiquées pour un appel à Paris :

| | Prix / heure de connexion |
|---|---|

| Appel de : | Tarif plein | 50% | 65% |
|---|---|---|---|
| Paris | 14,84 F | 7,42 F | 5,20 F |
| Proche Banlieue | 22,26 F | 11,13 F | 7,79 F |
| Province | 127,20 F | 63,60 F | 44,52 F |

Prix de l'unité Télécom : 0,742 FF.

# 12 Envisager d'écrire des applications

Vous avez sûrement plein d'idées d'applications intéressantes, mais envisager de contrôler une jonction série vous fait peur ? Voici quelques petites indications.

## 12.1 Et si c'était simple ?

Sous Unix, donc sous Linux, les seuls objets manipulés lors des entrées/sorties sont les *fichiers*. (Tiens, au fait, on aurait peut-être dû commencer par là ! Bon, vous le saviez déjà, ce n'est pas un cours Unix). La jonction série n'échappe pas à cette règle et le pilote vous la présente ainsi. Ici, il s'agit d'un (ou plutôt deux comme on le verra plus loin) fichier particulier, bien sûr, puisque se cache derrière un pilote (*driver*) en mode caractère, mais la façon de réaliser les entrées sorties est la même : **open, read, write, ioctl, close**. Il y a quand même quelques petites choses à savoir.

Le pilote série s'est enregistré, à l'initialisation du noyau, comme un *tty*. Il est donc géré comme un tty classique. Il apporte bien entendu quelques caractéristiques supplémentaires que nous verrons rapidement dans la section concernant la commande *ioctl*. Toujours est-il que si vous savez gérer un *tty* sous Unix (commande *stty*), vous saurez sans problème gérer ce pilote.

### 12.1.1 open, close

La méthode d'ouverture et de fermeture d'un port série est classique. Néanmoins un mécanisme particulier se cache derrière la primitive **open** qui est rapidement décrit à la section 12.2 (Appels entrants (Dial-in) et appels sortants (Call-out) sous Linux).

### 12.1.2 read, write

Rien à signaler de particulier.

### 12.1.3 ioctl

Le contrôle de tout système d'entrée/sortie se fait avec la commande système *ioctl*. Elle est souvent masquée par des commandes de haut niveau (setserial, stty, modemstat, ...), mais elle est leur moteur.

Selon que vous souhaitez vous adresser aux fonctionnalités classiques d'un *tty* ou aux fonctions spécialisées du pilote série, vous utiliserez deux sous-ensembles de commandes différentes. Vous les retrouverez dans le fichier à inclure *linux/termios.h*. Décrivons-les rapidement (on déborde un tout petit peu du sujet `:-)`) :

**TCGETS - TCSETS**

et quelques dérivés avec WAIT, FLUSH... Elles permettent de récupérer (resp. positionner) les attributs standard *tty* dans une structure *termios* (voir le fichier *linux/termios.h*)

**TIOCSTTY - TIOCNOTTY**

permettent de définir (resp. annuler) une session d'utilisation du *tty*. Ceci est visible car, entre autres choses, une conséquence est le changement des droits du fichier correspondant

```
Avant:
        crw-rw-rw-   1 root     tty        4,  64 Nov 26 20:47 ttyS0
Apres:
        crw--w--w-   1 root     root       4,  64 Nov 26 20:49 ttyS0
```

**TCFLSH et compagnie**

positionnement d'indicateurs (voir la commande stty)

**TIOCGSERIAL - TIOCSSERIAL**

permettent de récupérer (resp. positionner) les informations générales dans une (resp. à partir d'une) structure *serial_struct* (voir le fichier *linux/serial.h*) : le type de port série, la ligne, le port, l'irq le port utilisé... ni plus ni moins ce que fait *setserial*.

**TIOCMGET - TIOCMSET**

permettent de récupérer (resp. positionner) les informations plus spécifiques à la jonction proprement dite (dans un entier, sous forme de bits positionnés selon que l'indicateur correspondant est vrai ou faux) :

```
+--------------------------------------------------------------------------------------------+
|   //     DSR      RNG       CAR        CTS        //   //    RTS          DTR         //|
|  //   (Data Set  (Ring)  (Carrier)  (Clear To    //   //   (Request   (Data Terminal   // |
| //      Ready)                        Send)     //   //     To Send)      Ready)      //  |
+------+----------+-------+----------+-----------+----+----+-----------+---------------+----+
 31    9     8         7         6          5        4    3        2          1            0
```

## 12.2 Appels entrants (Dial-in) et appels sortants (Call-out) sous Linux

### 12.2.1 Introduction

Le pilote série du noyau de Linux propose de gérer un même port série à la fois pour les appels entrants et pour les appels sortants exploitant ainsi pleinement leur caractéristique bi-directionnelle. Il offre donc à l'utilisateur deux types de fichiers :

- `/dev/ttyS<n>` : sont généralement utilisés en entrée ;
- `/dev/cua<n>` : sont généralement utilisés en sortie.

### 12.2.2 Gestion

Chaque port série est enregistré deux fois auprès du gérant *tty* : une fois en mode *entrée* (ttyS, majeur 4) et une fois en mode *sortie* (cua, majeur 5). Voyons rapidement comment le pilote gère ensuite les "deux" ports :

- les ports séries `/dev/cua` sont gérés en mode *non-bloquant*. Leur ouverture n'est possible que si la ligne `/dev/ttyS` correspondante n'est pas ouverte et active (sinon `errno` retourne EBUSY) ;
- les ports séries `/dev/ttyS` sont gérés en mode bloquant ou non-bloquant, c'est donc un peu plus compliqué. Si l'indicateur CLOCAL est positionné, l'ouverture en mode **bloquant** est effective si la ligne `/dev/cua` est libre. Si l'indicateur CLOCAL n'est pas positionné, elle est effective si les deux conditions suivantes sont réunies :
  - la ligne est libre (le `/dev/cua` correspondant n'est pas utilisé),
  - la porteuse (circuit 109) a été détectée.

  Dans ce mode et pendant que l'ouverture est bloquée, la ligne n'est pas occupée, ce qui signifie qu'une application peut toujours effectuer un appel sortant. Si le port est en cours de fermeture, l'ouverture échoue (EAGAIN).

  L'ouverture en mode non-bloquant, quant à elle, est effective si le port n'est pas déjà ouvert et actif (sinon `errno` retourne EBUSY)[12].

C'est le mode qu'utilisent beaucoup d'applications comme *getty* qui souhaitent dans un premier temps initialiser la ligne (pour éviter des instabilités liées aux connexions précédentes) voire ensuite pour initialiser l'équipement (modem). Elles ne s'intéressent qu'au fait que la ligne soit occupée, en fermeture ou libre. Si celle-ci n'est pas libre, l'application se termine et le mécanisme du *respawn* se charge de les relancer.

Pour la gestion de la connexion proprement dite, l'application *getty* (pour prendre un exemple courant) ouvre la ligne, soit en mode bloquant si vous laissez le modem en réponse automatique, soit en mode non-bloquant si vous souhaitez qu'elle gère activement la connexion. Elle est alors en attente bloquante en lecture (sur *read*).

# 13 Quelques applications intéressantes

## 13.1 Un numéroteur

Cette idée va faire plaisir à *Xavier CAZIN* : c'est la sienne. Le mieux est de le laisser parler :

> En fait, je trouverais très utile de cliquer sur un bouton pour appeler une personne retrouvée dans une base de données par exemple. Surtout si elle se trouve à l'étranger (minimum 12 chiffres depuis ici). Donc, ce que j'aimerais, ce sont les renseignements nécessaires au programmeur pour pouvoir construire un frontal qui demande au modem de composer le numéro choisi, puis affiche un message (si le poste n'est pas occupé) du style "Le téléphone sonne, prenez le combiné". Ces petites choses toutes simples demandent de comprendre ce que signifie prendre la ligne et la relâcher, pour un modem.

Excellent exercice. Alors avec tout ce que l'on vient de dire, au travail `:-)`. En laissant de côté la partie base de données, l'algorithme à utiliser correspond à *l'organigramme d'un appel* donné à titre d'information au chapitre 4 (Le modem), auquel il faut rajouter :

- au préalable, ouvrir le port série (`/dev/cua<n>`), le configurer. Le mieux est de prendre exemple sur ce que fait *getty* ;

[12]Les applications utilisent de plus le mécanisme des fichiers de verrouillage garantissant l'unicité d'utilisation de la ressource.

- envoyer ensuite les commandes AT d'initialisation puis de numérotation (utiliser la fonction chat()) qui fait ceci très bien ;
- enfin se mettre en lecture sur le port afin de détecter les messages émis par le modem. Le seul problème ici est qu'il sera difficile d'attendre certains d'entre-eux (NO ANSWER, TIMEOUT) étant donné qu'il faut prévenir l'utilisateur qu'il peut décrocher.

Par contre les messages NO DIALTONE (modem pas branché), et BUSY sont fort intéressants et permettent d'informer l'utilisateur immédiatement.

## 13.2 modemstat et compagnie

Vous trouverez sur les *serveurs ftp* habituels quelques petits programmes permettant d'afficher l'état de la jonction. Vous pourrez vous intéresser particulièrement à la façon de récupérer les informations (plutôt que sur l'interface utilisateur qui est l'exemple même de ce qu'il ne faut pas faire).

```
/pub/linux/sunsite/system/Serial/modem-stats-1.0.tar.gz
/pub/linux/sunsite/system/Serial/statserial-1.1.tar.gz
/pub/linux/sunsite/system/Serial/modemstat-0.2.tgz
```

Un bon point pour statserial aussi simple que bien présenté, en mode texte.

## 13.3 Un détecteur de signal 109 (CD)

Lu dans *comp.os.linux.development.apps* cette demande

> Existe-t-il un utilitaire que je pourrais utiliser dans un *shell-script* et retournant une valeur, disons 1, si le signal *Détection de porteuse* est monté et 0 sinon ?

Le message ne dit pas quel en serait l'usage mais peu importe c'est un excellent exemple d'utilitaire assez facile à réaliser. Voici d'ailleurs un exemple de code source possible :

```
------------------------- debut de carrier.c -------------------------
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <termios.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include <stdio.h>

main(int argc, char *argv[])
{
  char *whoami, *device;
  int fd;
  int modem_bits;
```

```
  whoami = (whoami = strrchr(argv[0], '/')) ? whoami + 1 : argv[0];

  if (argc != 2) {
    fprintf(stderr, "Usage: %s device-file\n", whoami);
    return(EXIT_FAILURE);
  }

  device = argv[1];

  if ((fd = open(device, O_RDONLY | O_NDELAY)) < 0) {
    fprintf(stderr, "%s: error opening \"%s\": %s\n", whoami, device,
            strerror(errno));
    return(EXIT_FAILURE);
  }

  if (ioctl(fd, TIOCMGET, & modem_bits) < 0) {
    fprintf(stderr, "%s: error getting modem line statuses for \"%s\": %s\n",
            whoami, device, strerror(errno));
    return(EXIT_FAILURE);
  }

  if (modem_bits & TIOCM_CAR) {
    printf("1\n");
    return(EXIT_SUCCESS);
  }

  printf("0\n");
  return(EXIT_FAILURE);
}
------------------------- fin de carrier.c -------------------------
```

On pourra ensuite l'utiliser dans un *shell* de la manière suivante :

```
if [ `carrier /dev/modem` -eq 1 ]; then
        ... choses a faire si la porteuse est detectee ...
else
        ... choses a faire s'il n'y a pas de porteuse ...
fi
```

# 14 Remerciements

Ce document n'aurait vu le jour sans un certain nombre de gens fort sympathiques et plein de bonnes idées appartenant pour la plupart à la mailing list de traduction :

René COUGNENC,\\ Xavier CAZIN,\\ Bernard CHOPPY,\\ François AUDIBERT\\ Eric DUMAS,\\ Pierre FICHEUX, \\ Hervé MIGNOT, \\ Pierre VASSELLERIE, \\ Jacques LAVIGNOTTE, \\ et tous les autres.