

Connexion des Ordinosaires à Internet – HOWTO

v1.00

Passerelle Série/Ethernet pour connexion d'un IBM PC XT à Internet Mise en œuvre des protocoles PPP et SLIP

Emmanuel DA PIEDADE - 01/09/2008

Pour tout commentaire/suggestion/erreur, vous pouvez me contacter à l'adresse suivante :
emmanuel.dapedade A.T gmail D.O.T com

Ce document décrit la connexion d'un Compatible IBM PC XT à Internet via une passerelle Série/Ethernet OpenBSD Cette méthode peut néanmoins s'appliquer aux ordinosaires possédant un port série ainsi qu'un client PPP/SLIP.

I) Introduction

Qu'y a t-il de plus banal aujourd'hui que la connexion d'un PC à Internet ?

Connecter nos ordinateurs au réseau des réseaux est pratique, voire indispensable pour beaucoup d'entre nous. Il fut un temps, vers le milieu des années 90, où cet acte n'avait rien d'anodin. Pas de Wifi, ni routeurs, ni DHCP permettant une configuration automatique et facile pour les particuliers. C'était l'époque des modems sur port série, des coupleurs acoustiques, des commandes AT Hayes, ainsi que celle des protocoles SLIP et PPP. La configuration d'une connexion n'était pas aussi simple qu'elle peut l'être aujourd'hui, depuis la démocratisation d'Internet en cette toute fin du millénaire.

Pour nous collectionneurs et amateurs de vieilles machines, il est intéressant de pouvoir connecter nos ordinosaires à Internet. Même si à l'heure du Web 2.0 il est difficile d'exploiter convenablement ce dernier avec les faibles ressources dont disposent nos machines favorites, aucun passionné ne restera insensible devant un ordinateur de plus de 25 ans connecté à l'Internet moderne.

Mais comment mettre en œuvre une telle connexion sur des machines vieilles d'un quart de siècle ? Quelle est la meilleure méthode ?

Tout d'abord, il faut savoir que toutes les machines n'ont pas les mêmes possibilités en matière de communication. Cela est surtout conditionné par les ports d'entrées/sorties disponibles.

Rares sont celles disposant d'origine d'une interface Ethernet et d'une pile TCP/IP (obligatoire), moyen le plus pratique de les connecter à Internet de nos jours. Certains amateurs particulièrement doués ont mis au point des cartes d'extension permettant d'équiper des machines 8 bits de telles interfaces, mais la plupart du temps il faudra se contenter des ports de communication fournis d'origine. Le plus pratique d'entre eux est sans conteste le port série (norme RS-232C-EIA-232, ou V24 du CCITT), qui a régné en maître depuis son invention dans les années 60 jusqu'au début des années 2000, date à laquelle la baisse des prix des équipements réseaux Ethernet et

l'avènement de l'ADSL (sans oublier le succès du port USB) ont permis la démocratisation de ces derniers au détriment des modems série RTC (mais aussi des autres équipements série).

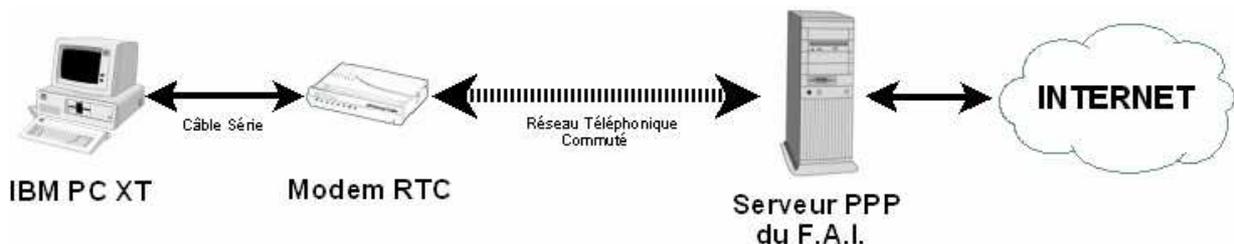
Encore aujourd'hui, l'équipement le plus basique pour connecter un ordinateur à Internet reste le modem RTC (Réseau Téléphonique Commuté – Nous pouvons même ajouter qu'il est le SEUL moyen disponible dans les zones « blanches » sans accès à l'ADSL)

Par exemple, un Atari 520ST, équipé d'un modem série et de la suite logicielle Sting peut se connecter à Internet par ce moyen. Mais depuis peu, avec le déploiement (inhérent aux offres triple-play des FAI) de la technologie VOIP (Voice Over IP) - par laquelle sont effectués un nombre grandissant des appels téléphoniques aujourd'hui - rend l'utilisation de cette méthode impossible. En effet, le VOIP, basé sur les datagrammes UDP, ne permet pas une qualité de service suffisante pour faire passer le protocole PPP utilisé pour une connexion via Modem RTC.

Comment donc connecter nos ordinosaires via leur port série s'il n'est plus possible d'utiliser les connexions RTC encore proposées par certains FAI ?

Il existe une méthode consistant à employer des techniques similaires à celle utilisées par les FAI justement, à savoir les serveurs PPP (encore utilisés aujourd'hui) et SLIP (complètement obsolète, quoi que, pas tant que ça ;-).

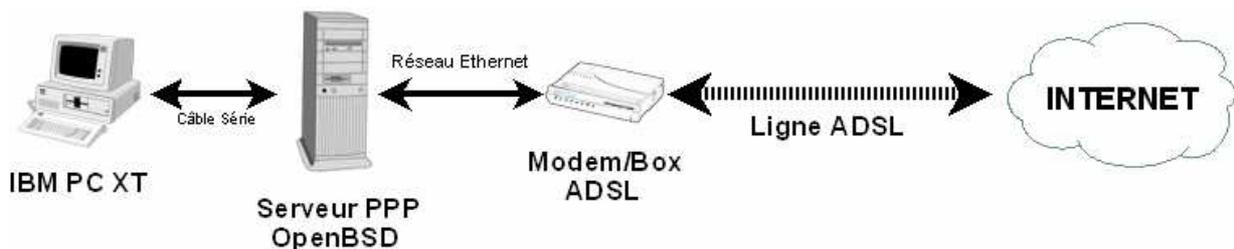
Voici un diagramme explicatif :



Ici, dans le cas classique, le PC de l'utilisateur se connecte au serveur PPP du FAI en passant par le modem qui convertit les signaux numériques de l'ordinateur en signaux analogiques sur le réseau téléphonique commuté (RTC), et inversement. Or nous savons que le VOIP ne permet plus l'utilisation des modems RTC. Comment contourner ce problème ?

Les FAI aujourd'hui proposent un modem/routeur (multimédia qui plus est) appelé Box qui permet la connexion de tous les appareils informatiques à Internet. Nous allons connecter notre ordinosaire indirectement à Internet via le routeur "Box" ADSL.

Voici un second diagramme pour comprendre le procédé:



Notre ordino-saure ne disposant que d'un port série, les seuls protocoles utilisables sont donc le PPP et le SLIP. Nous sommes donc obligés d'utiliser ces protocoles pour nous connecter. Le serveur PPP du F.A.I. n'étant plus accessible via le VOIP, nous allons connecter notre IBM PC XT à un serveur PPP maison, qui se chargera de rediriger les communications vers le Routeur/BOX ADSL, qui à son tour se chargera des les envoyer vers Internet.

La Box/Routeur du F.A.I. met à disposition de l'utilisateur un réseau local, par l'intermédiaire d'un switch, sur lequel viennent se connecter les équipements informatiques filaires (par opposition aux équipements Wi-Fi qui se connectent à la Box par l'intermédiaire d'un Point d'Accès Wi-Fi).

Notre serveur PPP/SLIP, que nous appellerons Passerelle Série/Ethernet, sera donc connecté d'un coté au réseau local sur le switch du routeur BOX/ADSL, et de l'autre coté à l'ordino-saure via son port série.

II) Mise en oeuvre

Pré requis :

- Connaissances de base en réseaux TCP/IP (configuration IP, Gateway, DNS ...)
- Connaissances de base en UNIX
- Connaissances de base des liaisons série RS232C (bauds, parités, câblage ...)

Matériel nécessaire :

- Un ordino-saure équipé d'un port série et d'une pile TCP/IP PPP ou SLIP
- Une connexion Internet avec routeur/Box
- Un PC comportant au minimum une carte réseau et un port série faisant office de passerelle
- Un câble null-modem (3 ou 7 fils)

Je prendrais ici comme exemple la connexion à Internet d'un compatible IBM PC XT avec 640ko de RAM.

La passerelle peut être constituée de n'importe quel PC capable de faire tourner un OpenBSD récent, équipé au minimum d'une carte Ethernet pour la connexion au réseau local et d'un port série pour y connecter notre ordino-saure.

Notez qu'il est aussi possible d'utiliser ici n'importe quelle version de Linux dont le noyau aura été compilé avec le support de PPP et SLIP, mais nous n'évoquerons ici que les lignes de commande et les fichiers de configuration propres à OpenBSD.

1) Préparation de l'ordino-saure et choix du protocole

Notre ordino-saure doit remplir plusieurs critères :

- Posséder un port série RS232C
- Posséder un client PPP et/ou SLIP, en plus des traditionnels logiciels Internet (newsreader, browser Web etc...)

Vient ensuite la question du choix du protocole pour la connexion série : PPP ou SLIP. Quel est le meilleur ?

PPP est un protocole robuste, flexible et standard qui fonctionne au niveau de la couche de liaison (Niveau 2) du modèle OSI. Il supporte plusieurs protocoles réseaux en plus d'IP (Novell IPX, DECNet ...), la négociation des paramètres de liaison et IP, l'authentification (CHAP, PAP), et la détection d'erreurs de communication.

SLIP est quant à lui un protocole beaucoup plus léger qui encapsule simplement le protocole IP. Il ne fournit aucun contrôle d'erreur ou d'adresse.

Il est adapté aux ordinateurs les plus modestes, possédant peu de RAM. Mais bien évidemment, sa simplicité le rend nettement moins fiable que PPP, car il ne possède aucun mécanisme de détection des erreurs. Néanmoins, son protocole plus léger lui permet d'être plus efficace que PPP (à caractéristiques de lignes équivalentes) en terme de vitesse de transfert TCP/IP. Il existe une version spéciale de SLIP, appelée CSLIP, qui permet, au prix d'une consommation CPU un peu plus élevée, de gagner en vitesse grâce à la compression des en-têtes TCP/IP Van Jacobson. Si vous avez le choix, optez pour PPP qui est le plus fiable des deux, malgré sa consommation CPU et mémoire plus importante.

Faites donc votre choix (PPP, SLIP ou CSLIP) en fonction des protocoles disponibles pour votre ordinateur.

En ce qui concerne le compatible IBM PC XT utilisé en exemple, voici les « packet drivers » les plus courants :

PPP : PPPD v0.6 (un port du daemon PPPD Unix pour MSDOS par Antonio Lopez Molero)

SLIP : SLIPPER v1.3, SLIPKT v0.56t et ETHERSLIP v11.7

CSLIP : CSLIPPER v1.3 et CSLPKT v0.54t

Vous pourrez les trouver à cette adresse : ADRESSE

2) Préparation de la passerelle OpenBSD (serveur PPP/SLIP)

Nous allons ici nous atteler à la configuration de notre passerelle qui fera le lien entre le réseau local et la liaison série de l'ordinateur.

Ce HOWTO ayant pour pré requis un minimum de connaissances en UNIX, je vous renvoie vers l'excellent guide officiel pour ce qui concerne l'installation d'OpenBSD sur le PC :

<http://www.openbsd.org/faq/fr/faq4.html>

L'installation par défaut est amplement suffisante. Il vous faudra peut être installer le paquetage Minicom pour debugger votre liaison série, mais cela reste facultatif. Si vous n'avez pas de PC disponible pour mettre en place votre passerelle (ou bien que la place vous manque), vous pouvez en émuler un en utilisant la virtualisation avec VMWare. C'est extrêmement efficace et pratique. Configurez donc le réseau à l'installation, il n'y a rien de spécial à faire pour peu que vous soyez

en DHCP. Dès lors que votre PC sous OpenBSD sera connecté à Internet via votre Routeur/Box, nous passerons à la configuration des serveurs PPP/SLIP.

3) Choix du câble série

C'est le câble série qui assurera la communication entre notre ordinozure et la passerelle. Il comporte à ses extrémités des connecteurs DB9M ou DB25M. Les plus simples sont constitués de 3 fils (TX, RX et masse). Ces derniers sont suffisants, mais il ne vous sera pas possible d'utiliser le contrôle de flux matériel basé sur les signaux CTS/DTS.

Les câbles permettant ce fameux « hardware handshaking » comportent 7 fils.

Avec SLIP et PPP, le contrôle de flux matériel peut apporter un léger plus surtout avec des vitesses élevées et/ou des machines lentes.

Vous pouvez vous fabriquer votre propre câble, en fonction des connecteurs (DB9 ou DB25), de la longueur qu'il vous conviendra, en suivant ces instructions :

<http://www.aurelienr.com/electronique/nullmodem.htm>

4) Test de la ligne série

Il nous faut maintenant tester le bon fonctionnement de la ligne série. Reliez l'ordinozure à la passerelle OpenBSD via leurs ports série. La méthode idéale est de démarrer de chaque côté un programme d'émulation de terminal. Les ports série doivent être configurés de manière identique sur les deux machines, par exemple en 9600 N 8 1 (9600 bauds, pas de parité, 8 bits de données, 1 bit de stop). Tout ce qui est tapé sur le clavier de l'ordinozure doit s'afficher sur l'écran de la passerelle, et inversement. En cas de mauvais fonctionnement, vérifiez les paramètres des ports, mais aussi le câble avec un testeur de continuité, pour détecter d'éventuels mauvais contact ou erreurs de câblage.

5) Connexion de l'ordinozure à la Passerelle

Tous les éléments nécessaires à la connexion de notre ordinozure à Internet sont prêts.

Nous allons maintenant configurer et activer la connexion PPP/SLIP à proprement parler.

Les commandes qui suivent sont spécifiques à la connexion d'un compatible IBM PC XT à la passerelle OpenBSD, mais il est possible d'appliquer cette procédure à d'autres ordinozures en adaptant ces dernières.

Une passerelle établit un lien (logiciel) entre deux interfaces réseaux.

La première interface réseau est celle connectée au réseau local et à notre Box/Routeur.

Pour lister les interfaces disponibles, tapez dans une console :

ifconfig

Dans l'exemple ci-dessous, notre interface Ethernet s'appelle *pcn0*.

Il nous faut maintenant notre deuxième interface. Celle-ci communiquera directement avec l'ordinozure via le port série de notre PC passerelle. Pour une interface PPP, le nom de l'interface sera *ppp0*, pour une interface SLIP, *sl0*.

En fonction du protocole que vous choisirez d'utiliser (PPP, SLIP ou CSLIP), les commandes à saisir seront différentes. Reportez vous à la rubrique adéquate.

Mais avant d'aller plus loin, il nous faut activer la NAT si nous voulons que les paquets transitent d'une interface réseau à l'autre, ainsi que l'IP Forwarding.

Commençons par activer l'IP Forwarding avec la commande :

```
sysctl net.inet.ip.forwarding=1
```

On peut aussi l'activer automatiquement à chaque démarrage du système en ajoutant cette même ligne dans le fichier */etc/sysctl.conf* .

Maintenant, pour activer la NAT, nous allons modifier le fichier de configuration de l'excellent Firewall OpenBSD (Packet Filter, ou PF) : */etc/pf.conf*

Vous pouvez soit créer un nouveau fichier (à l'aide de votre éditeur préféré), ou bien modifier celui existant en décommentant/modifiant les lignes adéquates.

Notre fichier */etc/pf.conf* doit contenir :

```
ext_if="pcn0"  
int_if="ppp0"  
nat on $ext_if from !($ext_if) -> ($ext_if:0)
```

Maintenant, nous allons démarrer PF pour mettre en service la NAT :

```
pfctl -e
```

Voici quelques commandes pratiques pour gérer PF :

```
pfctl -e (enable)  
pfctl -d (disable)  
pfctl -f /etc/pf.conf (indique le fichier de conf. à charger)  
pfctl -s state (liste les sessions NAT en cours)  
pfctl -k host (supprime les sessions NAT relatives à l'hôte host)
```

Notez aussi que vous pouvez automatiser le démarrage de PF en mettant la ligne **PF=YES** dans */etc/rc.conf.local* .

Maintenant que nous pouvons faire transiter les paquets entre l'interface Ethernet et l'interface série, il nous faut créer cette dernière. Le tableau suivant récapitule les deux lignes de commande à utiliser sous OpenBSD, pour chacune des combinaisons Protocole/Contrôle de flux possibles. La première crée l'interface elle-même. La seconde initie la connexion sur la ligne série.

Pour cet exemple, nous utiliserons le premier port série (COM1, tty00 ou encore cua00) ainsi qu'une vitesse de 9600 bauds.

Saisissez les deux commandes qui correspondent au protocole que vous avez choisi :

Protocole / Contrôle de Flux	Passerelle OpenBSD
PPP sans handshaking	<i>ifconfig ppp0 up</i>

	<i>pppd debug local noauth persist 10.0.0.1 10.0.0.2 /dev/tty00 9600</i>
PPP + RTS/CTS	<i>ifconfig ppp0 up pppd debug local crtscts noauth persist 10.0.0.1 10.0.0.2 /dev/tty00 9600</i>
PPP + XON/XOFF	<i>ifconfig ppp0 up pppd debug local xonxoff noauth persist 10.0.0.1 10.0.0.2 /dev/tty00 9600</i>
SLIP sans handshaking	<i>ifconfig sl0 10.0.0.1 10.0.0.2 -link0 -link1 -link2 up slattach -s 9600 /dev/cua00</i>
SLIP + RTS/CTS	<i>ifconfig sl0 10.0.0.1 10.0.0.2 -link0 -link1 -link2 up slattach -h -s 9600 /dev/cua00</i>
CSLIP sans handshaking	<i>ifconfig sl0 10.0.0.1 10.0.0.2 link0 -link1 -link2 up slattach -s 9600 /dev/cua00</i>
CSLIP + RTS/CTS	<i>ifconfig sl0 10.0.0.1 10.0.0.2 link0 -link1 -link2 up slattach -h -s 9600 /dev/cua00</i>

Voici quelques explications sur les paramètres utilisés pour les connexions PPP :

ppp0 est le nom de l'interface PPP.

debug active le mode debug, utile pour comprendre d'où viennent les problèmes de connexion.

local indique qu'il s'agit d'une connexion directe par câble série.

noauth désactive l'authentification.

persist rend la connexion persistante, en cas de déconnexion ou de time-out, pppd ne se terminera pas mais attendra une nouvelle connexion.

10.0.0.1 est l'adresse IP de notre passerelle BSD sur l'interface ppp0.

10.0.0.2 est l'adresse IP de la machine à l'autre extrémité de la ligne série, ici l'ordinateur.

/dev/tty00 est le device correspondant au port série à utiliser (ici l'équivalent de COM1 sous DOS/Windows).

9600 est la vitesse de communication en bauds.

crtscts (optionnel) active le contrôle de flux matériel RTS/CTS (nécessite un câble série complet).

xonxoff (optionnel) active le contrôle de flux logiciel XON/XOFF.

Les paramètres pour les connexions SLIP :

sl0 est le nom de l'interface

10.0.0.1 est l'adresse IP de notre machine BSD sur cette interface **sl0**

10.0.0.2 est l'adresse de la machine (ordinateur) connectée à l'autre extrémité de la ligne série

link0 active la compression d'en-têtes Van Jacobson (pour CSLIP), **-link0** la désactive

link1 ne laisse pas passer les paquets ICMP (ping), *-link1* les autorise
link2 active la compression automatiquement en cas de réception de paquets à en-têtes compressées, *-link2* la désactive

Pour *slattach*, *-s 9600* paramètre la vitesse de communication à 9600 bauds.

/dev/cua00 est le device correspondant au port série à utiliser (ici l'équivalent de COM1 sous DOS/Windows).

-h (optionnel) permet d'activer le contrôle de flux avec les signaux RTS/CTS (nécessite un câble série complet).

Notez que les devices de type */dev/ttyXX* et */dev/cuaXX* sous OpenBSD sont les homologues des ports COMx de DOS/Windows, mais chacun dans un mode d'accès différent.

Notre interface série a été initialisée et la communication démarrée du coté de la passerelle OpenBSD. Cette dernière est maintenant en attente d'une demande de connexion de la part de l'ordinosaure, dans notre exemple un IBM PC XT sous DOS. Saisissez maintenant la commande correspondant au protocole choisi (Vous trouverez les packet drivers PPPD, SLIPKT, CSLIPKT, et SLIPPER à l'adresse)

Protocole / Contrôle de Flux	IBM PC XT (Ordinosaure)
PPP sans handshaking	<i>pppd com1 9600 irq 4 local</i>
PPP + RTS/CTS	<i>pppd com1 9600 irq 4 local crtscts</i>
PPP + XON/XOFF	<i>pppd com1 9600 irq 4 local xonxoff</i>
SLIP sans handshaking	<i>SLIPKT /c 1 /s 9600 /v 0x60 ou SLIPPER com1 vec=60 baud=9600</i>
SLIP + RTS/CTS	<i>SLIPKT /c 1 /s 9600 /v 0x60 /h</i>
CSLIP sans handshaking	<i>CSLIPKT /c 1 /s 9600 /v 0x60 ou CSLIPPER com1 vec=60 baud=9600</i>
CSLIP + RTS/CTS	<i>CSLIPKT /c 1 /s 9600 /v 0x60 /h</i>

Remarques :

Le paramètre *60* ou encore *0x60* correspond au vecteur d'interruption de notre packet driver. Cette adresse nous sera utile notamment lorsque nous voudrions stopper la communication coté ordinosaure.

En cas de problème de connexion, vous pouvez saisir la commande *pppdd com1 9600 irq4 local debug* (Attention, *pppdd* avec deux *d* !) Cette version de de pppd vous donnera beaucoup plus d'indications sur l'origine du problème.

Pour certains packet drivers, il est nécessaire d'indiquer non seulement le port COM mais aussi l'IRQ utilisé. En général, les ports COM1 et COM3 utilisent l'IRQ 4, et les port COM2 et COM4 l'IRQ 3. Pour vous aider, des utilitaires comme MSD.EXE peuvent vous donner les IRQ de chaque port série.

Attention ! N'attendez pas trop longtemps, après avoir lancer pppd sur la passerelle, pour démarrer le packet driver coté ordino-saure, car durant une courte période, la connexion PPP est impossible. Lire le paragraphe 7) pour plus d'explications.

6) Et maintenant ?

Maintenant que nous avons établi une communication TCP/IP série entre l'ordino-saure et notre passerelle, nous pouvons utiliser les outils Internet MSDOS disponibles.

La meilleure pile TCP/IP DOS est sans conteste WATTCP. C'est en fait une librairie C utilisée pour le développement d'applications réseaux DOS dans les années 90.

Pour configurer TCP/IP sous DOS, il vous faut juste éditer ou créer le fichier WATTCP.CFG présent dans le répertoire de chaque application utilisant la librairie WATTCP.

Voici le fichier WATTCP.CFG utilisé dans notre exemple :

```
hostname=ibmpcxt
my_ip=10.0.0.2
netmask=255.255.255.0
gateway=10.0.0.1
nameserver=212.27.53.252
nameserver=212.27.54.252
```

Vous pourrez ainsi utiliser le browser web *bobcat* (basé sur lynx), ou encore l'excellente suite TCP/IP *Minuet* qui contient un lecteur de news, un client de messagerie (ne pouvant hélas plus fonctionner avec les serveurs SMTP modernes, en raison de l'implémentation de l'authentification SMTP pour les emails entrants), des clients FTP et Gopher, un browser Web ainsi que quelques outils réseaux (*ping*, très utile pour tester votre connexion, mais aussi *finger* et une sorte de *nslookup*).

Pour stopper les communications coté PC XT, il vous suffit d'utiliser la commande *TERMIN.COM* (que vous trouverez dans l'archive de PPPD) avec le vecteur d'interruption du packet driver pour paramètre :

```
TERMIN 0x60
```

Coté passerelle OpenBSD, taper *pkill -1 pppd* réinitialisera votre connexion PPP. *pppd* ,ne se terminera pas et attendra une nouvelle connexion.

Pour une connexion SLIP, il vous faudra faire un *pkill slattach* . Attention slattach se terminera aussitôt, et il vous sera nécessaire de le relancer en cas de nouvelle connexion SLIP avec l'ordino-saure.

7) Débug de la connexion RS232/PPP/SLIP - Conseils

Nous avons plusieurs outils à notre disposition pour vérifier le bon établissement de la connexion PPP/SLIP.

Un émulateur de terminal tel *Procomm* (MSDOS) ou encore *Minicom* (OpenBSD) vous permettra de vérifier le fonctionnement et le câblage de votre ligne RS232. Appliquez les mêmes paramètres sur les deux machines (comme par ex 9600 N 8 1), et tout ce que vous saisissez sur l'une sera retransmis à l'autre.

Une fois la liaison RS232 vérifiée, l'utilitaire *tcpdump*, disponible dans l'installation par défaut d'OpenBSD permet d'observer toutes les trames qui passent sur un interface réseau. Il est très efficace par exemple pour suivre les négociations PPP lors de la connexion.

Exemples d'utilisation :

tcpdump -X -i ppp0

tcpdump -X -i sl0

Voici quelques détails quant au comportement de *pppd*.

Lors de son lancement, il commence par envoyer pendant quelques minutes, des requêtes *LCP* de type *Configure-Request*, et attendra une éventuelle réponse de la part d'un client. En absence de cette dernière, il mettra fin à ses requêtes, et quelques secondes plus tard, se mettra simplement en écoute de requêtes en provenance d'un client. Durant ce temps mort, entre la fin de l'émission des requêtes *LCP Configure-Request*, et le début de l'écoute passive, il s'écoule quelques secondes pendant lesquelles aucune connexion n'est possible, c'est pourquoi il est important de ne pas initialiser de communication durant cette courte période. L'idéal étant de lancer le packet driver sur l'ordinateur juste après avoir démarré le serveur *pppd* sur la passerelle BSD.

Pour réinitialiser la communication avec *pppd*, envoyez le signal *SIGHUP* (1) au process correspondant avec la commande *pkill -1 pppd*.

Les signaux *SIGINT* (2) et *SIGTERM*(15) termineront *pppd* proprement. Si *pppd* refuse de quitter, utilisez alors le signal 3, le 6 ou en dernier recours le 9.

Pour vous aider à comprendre l'origine d'un problème, vous pouvez rendre certains process plus "parlant" en activant leur option *debug* ou *verbose*, mais aussi suivre l'établissement des connexions PPP et SLIP côté passerelle BSD en faisant un *tail -f /var/log/messages*.

Enfin, vous pouvez aussi à tout moment (comme toujours sous UNIX) consulter l'aide en ligne pour chaque commande utilisée dans cet How-to à l'aide de la commande *man* :

man pppd

man slattach

N'hésitez pas non plus à lire les fichiers de documentation fournis avec les packet drivers DOS qui regorgent d'informations utiles.