Les tunnels point à point

Présentation rapide

PPP, PPtP, PPPoE, PPPoA ... Autant d'acronymes barbares qui fleurissent sur le Net. Les Internautes câblés n'avaient jusqu'ici pas à se préoccuper de ces sujets de torture intellectuelle, nous devons malheureusement nous y plonger, FTCI ouvrant le bal.

Voici, rapidement présentés, les points que nous allons essayer de développer dans ce chapitre. Il s'adresse à tous les internautes connectés à "haut débit" (câble, mais aussi ADSL). D'ailleurs, la plupart des manips est faite sur une connexion ADSL.

La connexion Point à Point

Depuis longtemps l'informatique a besoin d'établir des connexions entre hôtes sur des supports qui ne sont pas des supports "réseau". Une liaison série RS 232 par exemple, comme celle qui reliait votre PC à votre Modem RTC, du temps où l'USB n'existait pas.

Connexion Internet via un modem RTC

Depuis le PC jusqu'à l'équipement du fournisseur d'accès, la connexion n'est pas de type "réseau" (du moins, jusqu'au concentrateur téléphonique). Internet nécessite l'usage du protocole TCP/IP. Il est donc nécessaire de mettre en dessous un autre protocole, capable de supporter TCP/IP, qui ne sera pas Ethernet (puisque la connexion n'est pas de type "réseau").

PPP (Point to Point Protocol) a été créé pour résoudre ce problème. PPP est capable de transporter sur une liaison série, sur une ligne téléphonique RTC par l'intermédiaire d'un modem, non seulement TCP/IP, mais tout protocole réseau comme IPX/SPX ou même NetBEUI.

Connexion Internet via le câble

Pour le câble, c'est différent. Nous avons une architecture réseau et il est parfaitement possible de transporter des trames Ethernet dessus (via ATM, qui arrive jusqu'au modem câble). Il n'est donc à priori pas nécessaire d'utiliser un protocole point à point, ce qui était d'ailleurs le cas jusqu'à présent.

Connexion Internet via ADSL

Bien que l'on utilise ici une ligne téléphonique, c'est ATM qui arrive jusqu'au modem de l'abonné. Nous sommes donc dans la même situation qu'avec le câble, il est possible de transporter des trames Ethernet jusque chez l'abonné, sans utilisation d'un protocole point à point. Les premières expérimentations de l'ADSL ont d'ailleurs eu lieu sur ce schéma. Les tunnels point à point ont été ajoutés pour permettre de partager le même support (boucle locale) entre plusieurs fournisseurs d'accès.

Câble ou ADSL sans support point à point

Dans cette configuration, tous les abonnés sont, dès lors qu'ils sont connectés, placés sur un même réseau physique. Il ne peut simplement y avoir qu'un seul fournisseur d'accès commun à tous les abonnés. Cette situation a été acceptée pour les câblo opérateurs, elle ne l'a pas été pour l'exploitation de l'ADSL.

Câble ou ADSL avec support point à point

Les deux technologies amènent un réseau ATM jusqu'au modem de l'abonné. ATM est un protocole réseau qui fonctionne en mode connecté. Autrement dit, il établit un chemin virtuel entre client et serveur pour réaliser la connexion. Mais TCP/IP fonctionne sur Ethernet. Finalement, les trames Ethernet sont transportées par ATM, qui devient complètement transparent pour l'utilisateur. Dans la pratique, avec un modem câble ou un modem ADSL relié au PC par un lien Ethernet, le PC se croit directement connecté à un réseau Ethernet.

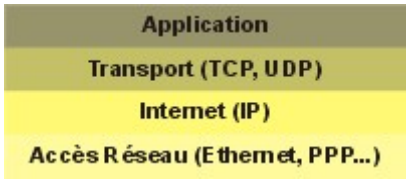
Si l'on ajoute au dessus d'Ethernet un protocole capable de faire du point à point entre le client et son fournisseur d'accès, on réalise une sorte de tunnel qui va permettre principalement d'utiliser le même "réseau" pour y faire cohabiter des clients de fournisseurs différents, la connexion se réalisant avec le bon fournisseur grâce à l'identifiant de connexion.

Cette stratégie a été mise en place sur ADSL au moment de sa mise en production, elle est en train de se mettre en place sur les réseaux câblés de FTC.

Plan du chapitre

IP sur quoi ?

Petit rappel



Peut-être est-il bon de rappeler ici le modèle DOD, celui sur lequel TCP/IP s'appuie.

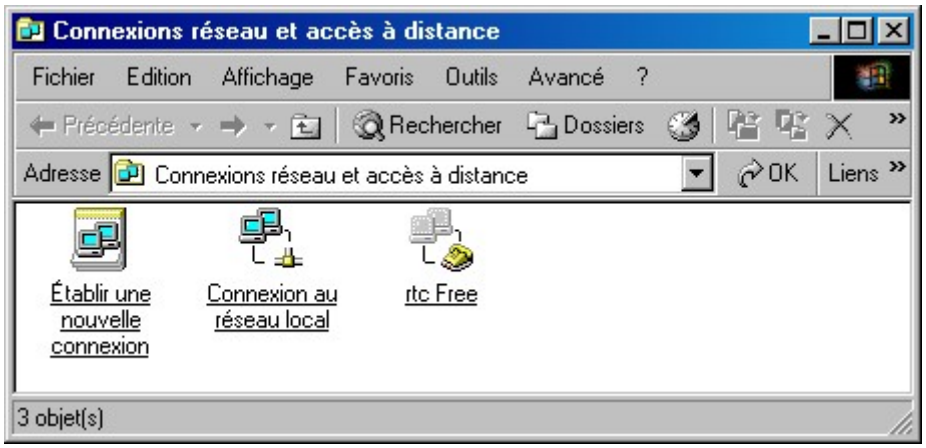
Dans le cas qui nous occupe, c'est principalement la couche d'accès réseau qui nous intéresse.

IP s'appuie avec aisance sur Ethernet. Mais il est tout à fait possible de remplacer Ethernet par quelque chose d'autre, pourvu que ce quelque chose procure une adresse MAC. Il faut que l'ARP fonctionne.

PPP, Point to Point Protocol

Nous savons tous qu'une connexion Internet fonctionne correctement si l'on ne dispose que d'un simple modem RTC (Réseau Téléphonique Commuté). Dans ce cas, il faut disposer d'un client PPP, capable d'établir une connexion PPP avec les équipements du FAI.

Sous Windows, ce client apparaît comme un "client d'accès distant".

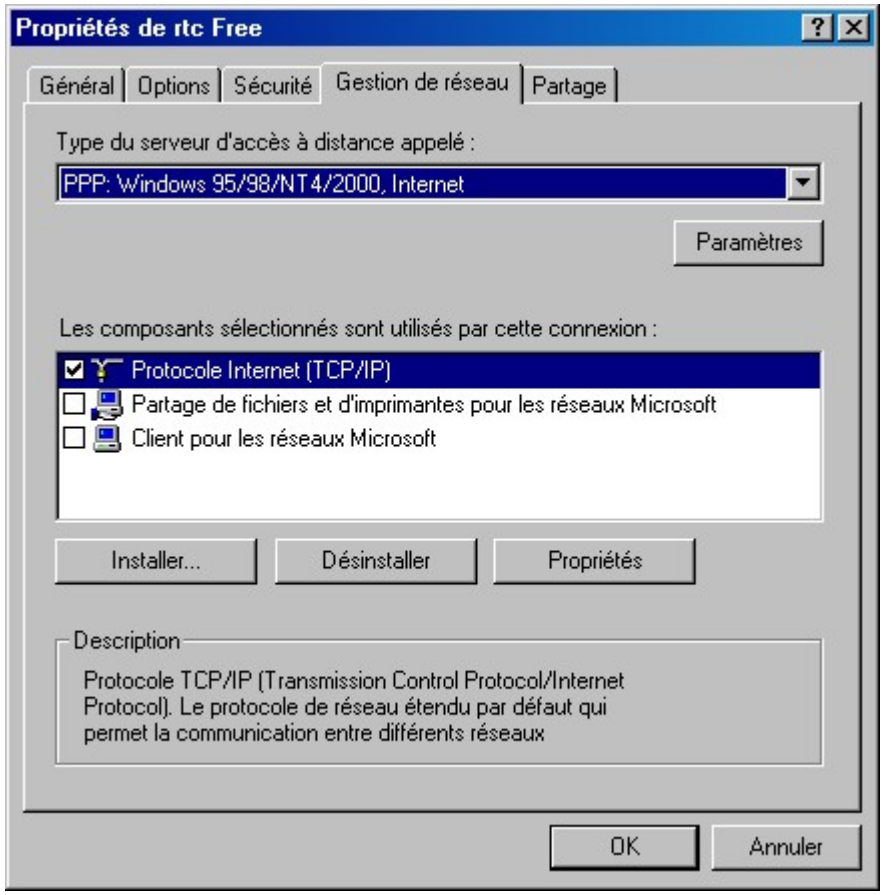


Voici typiquement ce que l'on voit sous WIndows 2000 par exemple, lorsque l'on dispose à la fois d'une connexion de type "réseau local" et une connexion par modem RTC (ici "rtc Free")

Si l'on développe les propriétés de cette connexion RTC, voici ce que l'on voit :

Un client PPP, capable de se connecter à des serveurs d'accès distants de type Windows 95/98/NT4/2000 (et XP bien sûr), mais aussi Internet.

Quelques tribulations dans cette fenêtre montreront que l'on dispose ici de toutes les possibilités de TCP/IP.

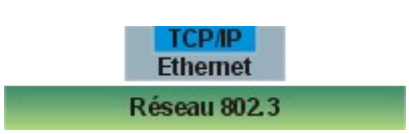


L'aspect de ces fenêtre différera suivant la version de Windows, mais vous retrouverez les mêmes fonctionnalités sur toutes les versions, à partir de Windows 95.

Tout ceci pour bien montrer que TCP/IP peut sans problèmes s'appuyer sur PPP plutôt que sur Ethernet.

Et l'adresse MAC ?

L'adresse MAC, normalement, est inscrite en "dur" dans l'interface Ethernet. Dans le cas de PPP, il n'y a bien entendu pas d'adresse MAC écrite en "dur" sur votre machine, le serveur d'accès distant va combler cette lacune (service proxy ARP).



A gauche, IP s'appuie sur Ethernet qui utilise un support de type 802.3

A droite, IP s'appuie sur PPP qui utilise (par exemple) un support de type liaison série RS 232

Remarques diverses

Ethernet et PPP n'ont pas grand chose en commun, si ce n'est qu'ils peuvent supporter tous les deux IP.

Ethernet

Nous l'avons largement vu par ailleurs sur ce site, Ethernet est un support "communautaire". les datagrammes circulent sur tout le réseau physique, tout le monde peut les voir passer, c'est juste la couche d'accès au réseau qui va, suivant l'adresse MAC du destinataire, décider de remonter ou non le datagramme aux couches supérieures. Cette technologie est très majoritairement utilisée sur les réseaux locaux.

PPP

Nous sommes en mode connecté point à point. Lorsque l'on utilise un modem RTC, depuis son PC jusqu'à l'équipement du fournisseur de services on utilise une ligne qui relie physiquement les deux machines et elles seules (au moins virtuellement, comme nous le verrons plus loin).

ATM, l'accès réseau qui travaille dans l'ombre

Le réseau France Télécoms est un réseau ATM. C'est le même réseau qui sert aussi bien pour la téléphonie que le transport de données informatiques. ATM sait transporter des données de toutes natures. Autrement dit, lorsque vous communiquez entre un point A et un point B en utilisant le réseau FT, vous utilisez de l'ATM sans le savoir (et c'est tant mieux pour vous, ATM n'est pas une petite affaire).

- ATM sait transporter de la téléphonie numérisée. Si votre téléphone travaille avec un signal analogique dans une bande passante de 4 KHz sur la fameuse "boucle locale", ce signal est rapidement numérisé avant d'être véhiculé par le réseau ATM. Il sera à nouveau transformé en signal analogique avant d'être injecté dans la boucle locale de votre interlocuteur. (Même chemin tortueux pour les signaux des modems RTC).
- ATM sait aussi transporter des signaux vidéo numérisés, mais ce n'est pas notre propos.
- ATM sait transporter des données informatiques :
    - "Directement", il existe des interfaces ATM qui permettent de construire un réseau informatique purement ATM.
    - "Indirectement", en transportant des datagrammes issus d'autres accès réseaux comme PPP ou Ethernet. Ces trames sont alors transportées sur ATM par le biais d'une couche d'adaptation (AAL comme ATM Adaptation Layer). Dans ce cas, ATM est parfaitement transparent et l'utilisateur peut avoir l'impression qu'il est sur un réseau Ethernet, même si ce n'est pas vrai. C'est le cas actuellement pour les Internautes câblés.

ATM est un réseau dit "commuté". Bien que ce soit un vrai réseau, au sens où les équipements sont connectés en réseau, il s'établit un chemin virtuel entre source et cible et pour les données qui transitent, tout se passe comme s'il y avait un "fil" qui relie directement et uniquement source et cible.

S'il est bon de savoir qu'ATM est partout dans notre vie d'Internaute, il est encore meilleur de savoir que l'on n'a pas trop besoin de s'en soucier.

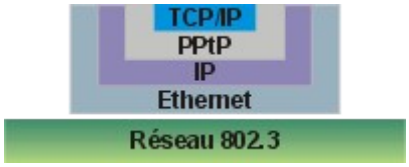
Et les tunnels ?

Jusqu'ici, tout est simple. Nous sommes en TCP/IP et, soit nous disposons d'un réseau 802.3 et utilisons Ethernet comme Accès réseau, soit nous disposons d'une ligne téléphonique et nous utilisons PPP comme accès réseau.

Seulement, voilà... Ethernet, pose beaucoup de problèmes du fait principalement que les données qui y circulent ne sont pas protégées des regards indiscrets. Les "renifleurs" (sniffers) en sont une brillante démonstration.

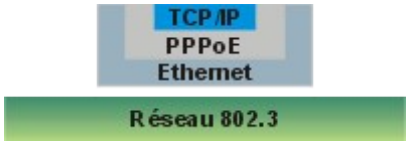
Il est donc devenu utile de créer au dessus d'Ethernet des connexions de type PPP. Ces connexions, bien que transportées par Ethernet apparaissent comme des liaisons point à point en exploitant les propriétés de PPP.

PPTP. (Point to Point Tunelling Protocol)



C'est un protocole qui a été développé par Microsoft. Son principe est de créer au dessus d'IP une liaison PPP dans laquelle on va refaire passer de l'IP. Deux couches IP l'une sur l'autre, si vous voulez. Ceci permet de créer des VPN (Virtual Private Network) au dessus d'un réseau public. Il est possible par ce biais de créer une sorte de LAN virtuel privé qui emprunte les voies de l'Internet.

PPPoE (Point to Point over Ethernet)



Ce protocole, également propriétaire au départ, a été conçu par la société RedBack. Son but, à peine différent, est de créer une liaison point à point au dessus d'Ethernet. Ici, il n'y a pas deux couches IP

Dans un cas comme dans l'autre, il s'agit de profiter des avantages d'une connexion point à point sur un accès réseau qui ne le propose pas. (On pourrait faire la chose nativement sur ATM, pas sur Ethernet).

Mais pourquoi des tunnels ?

A priori, ça permet de rendre les données plus confidentielles. Les tunnels peuvent être chiffrés et les connexions s'établissent par PPP et non par les moyens classiques d'IP directement sur Ethernet.

Pour un fournisseur de services Internet, l'intérêt principal est qu'il a une meilleure maîtrise des connexions de ses clients. De plus, le même réseau peut être partagé entre plusieurs fournisseurs, les clients établissant une liaison point à point avec leur FAI, comme par le RTC. C'est la solution qui a été retenue pour l'ADSL.

Et pourquoi sur le câble ?

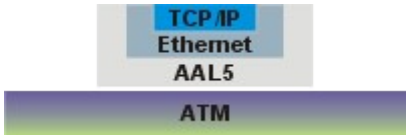
Demandez à votre FAI, lui le sait sûrement, moi je ne suis pas sûr de le savoir :)

PPPoE et le câble

La description qui suit s'appuie sur le cas de CâbleWanadoo.

Comme il a été dit plus haut, ATM apparaît ici pour mémoire, mais reste totalement transparent pour l'utilisateur.

Sans PPPoE



Nous sommes dans cette configuration :

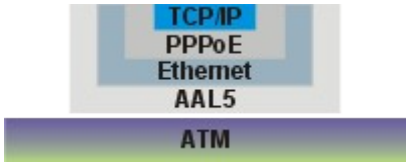
- IP accède au réseau par Ethernet
- Ethernet est transporté par ATM au moyen de la couche d'adaptation AAL5.

Pour le client connecté au Net par ce biais, tout se passe donc comme s'il était sur un réseau IP transporté par Ethernet, configuration identique à celle d'un LAN (Local Area Network).

Lors de la mise sous tension du poste client, si l'interface Ethernet est active, DHCP va faire son oeuvre et le client récupèrera automatiquement une adresse IP, une passerelle par défaut, un DNS, bref, tout ce qu'il faut pour qu'il soit connecté au réseau. Lorsque tout se passe bien, cette opération est complètement transparente pour l'utilisateur et il se trouve "de facto" connecté au Net.

Il n'y a aucune procédure de "login" avec nom d'utilisateur et mot de passe, comme ça se fait avec une connexion PPP sur RTC. L'utilisateur est en quelque sorte connecté de force. S'il veut s'isoler du réseau, il devra désactiver son interface réseau. En effet, une simple résiliation de bail ne suffira pas, la pile IP essayant à la première occasion d'obtenir un nouveau bail.

Avec PPPoE



Ici, nous sommes dans cette configuration :

- IP accède au réseau par PPP
- PPP est transporté par Ethernet.
- Ethernet est transporté par ATM au moyen de la couche d'adaptation AAL5.

Nous avons donc une couche de plus et IP se "voit" sur un transport PPP, exactement comme dans le cas d'une connexion par modem RTC, et non plus directement sur un transport Ethernet. Les analyses de trames qui suivent le montrent bien.

De ce fait, beaucoup de choses vont changer. (Imaginez que vous revenez en arrière, avec votre ancienne connexion RTC et, tout de même, quelques améliorations).

- Lors de la mise sous tension du client, rien ne devrait se produire concernant la connexion. (sauf si vous avez installé un script de connexion automatique).
- Un client DHCP ne fonctionnera pas. Ce protocole n'est utilisable que sur Ethernet. Votre adaptateur réseau n'a pas besoin d'être configuré en client DHCP. Il ne doit pas l'être, sinon votre pile IP va perdre du temps à chercher un serveur DHCP qu'elle ne trouvera forcément

pas.

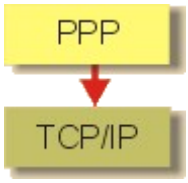
- Pour établir la connexion, il vous faudra initier une connexion PPP, comme avec votre bon vieux modem RTC (login, mot de passe) et c'est PPP qui récupérera alors votre IP, la passerelle par défaut et le DNS. Il n'y a plus de bail, les paramètres sont inchangés pour toute la durée de la session (24h maximum. vous serez automatiquement déconnecté si votre session dépasse 24h). Nous verrons cela dans le détail un peu plus loin.
- Vous pouvez vous isoler du réseau en fermant votre connexion PPP, exactement comme avec le modem RTC. En fait, si vous le souhaitez, vous pourrez travailler comme "avant" en n'établissant la connexion qu'à la demande et en la refermant automatiquement après une période d'inactivité définie. La différence, c'est que l'ouverture de la session PPP se fera en quelques secondes (3 dans l'exemple que nous verrons plus loin) et que le débit de votre modem restera ce qu'il était avant passage à PPPoE.
- Un HUB entre plusieurs PC et le Com21 ou le Thomson TCM290 ne vous permettra plus de partager la connexion. Tout ce que vous pourrez faire avec, c'est ouvrir une connexion PPP sur un et un seul PC à la fois. Normalement, votre FAI doit être capable de détecter l'emploi simultané de plusieurs logins identiques et comme vous n'en aurez qu'un...
  Il existe cependant toujours la possibilité de partager la connexion via un routeur NAT, hardware ou construit sur un PC dédié, par exemple sous Linux[1]. Si vous êtes dans ce cas, c'est le routeur qui va ouvrir la connexion PPP, les autres PC du réseau privé restant configurés comme ils l'étaient avant le basculement. Mais attention au MTU...

1 Masquerade : http://christian.caleca.free.fr/masquerade/index.html

PPPoE Installation

Ce qu'il faut d'abord comprendre

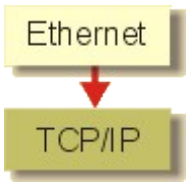
Le cas du modem RTC



Lorsque nous installons une connexion PPP, telle que celle que nous utilisons sur une liaison série avec notre modem RTC, Il n'existe pas à proprement parler d'interface physique.

Sous Windows, par exemple, nous installons le "client d'accès distant" que nous lions à notre modem, puis nous le configurons pour qu'il utilise TCP/IP pour la connexion Internet.

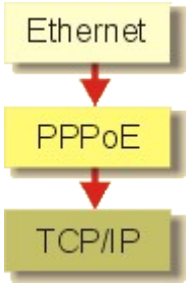
Le cas du réseau local Ethernet



Lorsque nous installons une interface Ethernet destinée à nous connecter sur un réseau local (ou sur le câble, avant passage à PPPoE), nous installons le "driver" de la carte réseau, puis nous y associons TCP/IP (et éventuellement, d'autres protocoles, comme IPX/SPX ou NetBEUI).

Nous configurons alors TCP/IP, soit "en dur" dans le cas d'un petit réseau local, soit de façon automatique via DHCP dans le cas d'un réseau plus complexe, le câble par exemple.

Le cas de PPPoE



Dans le cas de PPPoE, vous l'avez compris, il y a une couche de plus.

- Nous installons le "driver" de la carte réseau.
- Nous lions dessus le client PPPoE. Suivant la forme de ce client, il pourra apparaître comme un client d'accès distant (cas de ras pppoe) ou comme quelque chose de spécifique (cas d'Enternet), sans oublier le cas particulier de Windows XP qui intègre nativement PPPoE.
- TCP/IP va venir se lier à ce client de façon plus ou moins transparente, suivant le client

Notez bien qu'à aucun moment, TCP/IP n'a été directement lié à Ethernet, comme c'est le cas pour le réseau local.

Pour les câblés, le passage en PPPoE devrait donc s'assortir de deux manipulations distinctes :

- La suppression de la liaison entre Ethernet et TCP/IP
  Nous nous retrouvons alors avec une carte Ethernet "orpheline", sans aucun protocole greffé dessus.

- L'installation du client PPPoE
  Il sera lié à cette carte Ethernet d'une part et à TCP/IP d'autre part.

Ça, c'est la théorie. Parce que, suivant le client, ce sera plus ou moins vrai, comme nous allons le voir dans la suite.

Les clients PPPoE les plus courants

EnterNet 300 et WinPoET et les autres...

Ce sont les clients "officiels", distribués par FTC, aussi bien dans les kits Netissimo que Câble. Ce sont des logiciels commerciaux, édités respectivement par Efficient Networks, Fine Point Technologies[2], BeWan... Ce sont les seuls officiellement supportés par le service technique de FTC.

EnterNet 300 existe pour les plate-formes suivantes :

- Windows: 95, 98, Millenium, NT 4 sp3, 2000 et même XP à partir de la version 1.5c.
  Le cas de Windows XP est cependant particulier, XP propose nativement la gestion de PPPoE, un logiciel client tiers n'est donc pas nécessaire.
- Linux, à partir des noyaux 2.2, avec pppd version 2.3.11
- Macintosh: OS 7.6 à 9.x (je présente toutes mes excuses aux utilisateurs de Macintosh, j'ignore tout de ce monde et n'en parlerai donc pas d'avantage).

WinPoET n'existe que pour les plate-formes Windows. MacPoET existe pour Macintosh, mais n'est pas fourni dans le nouveau kit CâbleWanadoo.

Depuis la "positive génération", le kit semble avoir encore changé. J'ai eu la surprise de constater que le kit daté d'août 2002 propose encore un nouveau client, signé cette fois-ci par BeWan...

J'ai effectué de rapides tests de ce n^{eme} client sur Windows 98 SE, il fonctionne aussi... Pour ce qui est du kit lui-même, à part les couleurs, pas grand chose de changé, toujours IE 5.5 alors que vous pouvez disposer de IE6SP1, toujours cette fâcheuse habitude de reconfigurer Outlook Express...

Vous m'excuserez de ne pas décrire une procédure détaillée de ce nouveau client, j'avoue éprouver une certaine lassitude, à force...

Avantages

- Fournis gratuitement par FTC et les problèmes techniques sont gérés par la "hot line".
- Simples à installer et à configurer.

Inconvénients

- Logiciels commerciaux, les mises à jour ne sont pas gratuites et FTC ne distribuera pas forcément les mises à jour de façon régulière. Compte tenu de la succession des divers clients, il est même probable que vous aurez quelques difficultés à effectuer la moindre mise à jour.
- Ces logiciels nécessitent impérativement (à l'exception du petit dernier de BeWan) d'installer

2 Fine Point Technologies : http://www.finepoint.com/

une pile IP sur l'interface réseau connectée au modem. Si ce détail ne pose pas de problèmes avec un modem ADSL de type Alcatel SpeedTouch par exemple ou le nouveau modem câble eurodocsis TCM290 de Thomson, il risque fort d'en poser sur un réseau câblé, avec les Com21.

Ras PPPoE

C'est un client développé par un informaticien indépendant[3]. Il est gratuit pour un usage non commercial, et existe pour Windows 98, Millenium et 2000. NT4 et 95 n'étaient pas supportés au moment de mes tests, ce qui n'est plus le cas depuis. Raspppoe peut donc désormais être utilisé sur toutes les plate formes Windows.

Avantages

- Il est vraiment gratuit pour un usage non commercial, ses mises à jour également.
- Il "colle" un peu plus à l'architecture Windows que ne le fait EnterNet ou WinPoET. En effet, il utilise les services d'accès distants, comme avec un modem RTC, ce que ne font pas EnterNet ni WinPoET.
- Il ne nécessite pas de pile IP connectée à l'interface réseau.
- Il "pèse" beaucoup moins lourd, l'archive ZIP ne fait que quelques dizaines de Ko!

Inconvénients

- Risque de disparition d'un projet de développement supporté par un seul programmeur (mais rien n'interdit dans ce cas de choisir un autre client).
- Pas de support officiel de la part de FTI.
- Nécessite un logiciel supplémentaire[4] pour assurer une reconnexion automatique en cas de coupure.
- Pas de version pour MAC OS.

rp-pppoe

C'est un client libre uniquement pour Linux, Solaris et NetBSD, développé par Roaring Penguin[5]. A mon sens, il ne présente que des avantages.

Ce client existe sous forme de packages RPM pour RedHat et Mandrake (lisez tout de même les informations contenues dans le site de Roaring Penguin). Disponible également en fichier source "tarball", à compiler soi-même.

Ce paquetage propose tout ce qui est nécessaire pour configurer un client et même quelques outils supplémentaires pour tester des clients. Il nécessite la présence de pppd.

3 RAS PPPoE : http://www.raspppoe.com/
4 ADSL Autoconnect : http://www.adslautoconnect.net/
5 Roaring Penguin : http://www.roaringpenguin.com/pppoe/

Comment faire alors ?

Très simplement. Le plus délicat sera peut-être de choisir son client. Je ne vais pas le faire à votre place, à vous de voir. Voici tout de même mon avis personnel :

Vous utilisez une plate-forme Win32

- Vous n'êtes pas trop porté sur les subtilités du système. Vous souhaitez pouvoir bénéficier de la "hot line" et vous n'aimez pas "bricoler" sur votre machine: Préférez peut-être EnterNet, WinPoet ou BeWan, suivant le kit dont vous disposez.
- Vous aimez bien maîtriser le fonctionnement de votre système, vous en connaissez les rudiments et bricoler dans votre configuration ne vous fait pas trop peur : Préférez peut-être le couple ras pppoe et ADSL autoconnect.
  Si je devais connecter une machine Win32 sur une liaison haut débit, je choisirai sans doute cette solution.

Avant de choisir définitivement, pensez tout de même à ces points particuliers :

- Vous utilisez Internet Explorer 6.
  Le kit FTI vous donnera des sueurs froides. Vous aurez peut-être intérêt à installer le client PPPoE manuellement si vous avez une âme sensible.
- Vous utilisez Windows XP.
  Il y a un client PPPoE "natif", utilisez plutôt celui-ci.

Vous utilisez une plate-forme Linux

Dans ce cas, vous êtes probablement accoutumé à configurer votre système vous-même. Quant à la "hot line" sur cette plate-forme....

Vous pouvez, bien entendu, utiliser EnterNet, mais la licence est tellement incompatible avec celle de Linux que vous choisirez très probablement rp-pppoe, au moins pour des raisons idéologiques.

Pour vous aider...

Installer le Kit de façon automatique sur plate-forme Win32 ne pose pas de problèmes, il est même étonnant de constater à quel point cette procédure fonctionne bien (y compris d'ailleurs la procédure de désinstallation), mais à une condition, et elle est de taille : **Vous ne devez pas avoir au préalable installé Internet Explorer 6 !!!** Faute de quoi, nous allons le voir de près, il va falloir essuyer de sa part quelques comportements absurdes. Mais lorsque vous aurez besoin d'installer ce kit, vous aurez peut-être entre les mains une version corrigée.

Si, comme il est vivement recommandable de le faire, vous avez régulièrement effectué les mises à jour de votre système (Windows Update) et d'Internet Explorer, vous êtes justement dans le cas où le kit ne fonctionne pas correctement (du moins, semble ne pas fonctionner correctement).

Suivez maintenant le guide qui vous convient le mieux.

Compte tenu des multiples versions successives du kit, vous m'excuserez encore une fois de ne pas avoir développé à chaque fois le détail de l'installation du kit, qui, dans tous les cas, ne vous laisse aucune initiative lors de son installation.

Pour conserver des PDF de taille acceptables, ces procédures dans des fichiers externes.

- Installation automatique du kit (toutes plate-formes Win32) version "Enternet"
- EnterNet 300 sur Windows 98 SE
- EnterNet 300 sur Windows 2000
- Installation automatique du kit (Windows 2000) version "WinPoet"
- RAS PPPOE sur Windows 98 SE
- RAS PPPOE sur Windows 2000
- Le client PPPoE de Windows XP
- rp-pppoe pour Linux (Mandrake et Debian)

Un échange ICMP vu de près

La manip qui suit est destinée à éclaircir un peu les idées sur l'empilement des protocoles.

Elle est effectuée sur une machine Linux connectée en PPPoE sur un modem SpeedTouch Home (Accès ADSL).

- L'interface Ethernet connectée au Modem est Eth1. Le client rp-pppoe assure la connexion ppp au dessus. Ceci fait apparaître une interface supplémentaire : ppp0
  Autrement dit, ppp0 est une interface PPP qui fonctionne au dessus de Eth1
  Voici un extrait de ce que donne "ifconfig" :

```
[root@gw root]# ifconfig
....
eth1 Lien encap:Ethernet HWaddr 00:60:8C:50:F0:DF
inet adr:10.0.0.10 Bcast:10.0.0.255 Masque:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
....
ppp0 Lien encap:Protocole Point-à-Point
inet adr:217.128.147.4 P-t-P:217.128.147.1 Masque:255.255.255.255
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1492 Metric:1
....
```

- Deux "sniffers", l'un sur eth1 (couche Ethernet), l'autre sur ppp0 (couche PPPoE), vont observer simultanément un "ping" sur l'habituel ftp.oleane.net.

Voici ce que l'on obtient :

Sur Ethernet (Eth1)	Sur PPPoE (ppp0)
Frame 4 (106 on wire, 106 captured) Arrival Time: Nov 30, 2001 16:32:12.060281 Time delta from previous packet: 3.750210 seconds Time relative to first packet: 8.636959 seconds Frame Number: 4 Packet Length: 106 bytes Capture Length: 106 bytes **Ethernet II**	Frame 1 (84 on wire, 84 captured) Arrival Time: Nov 30, 2001 16:32:12.059992 Time delta from previous packet: 0.000000 seconds Time relative to first packet: 0.000000 seconds Frame Number: 1 Packet Length: 84 bytes Capture Length: 84 bytes **Raw packet data**

Sur Ethernet (Eth1)	Sur PPPoE (ppp0)

```
    Destination: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 218c
    Payload Length: 86
Point-to-Point Protocol
    Protocol: IP (0x0021)
Internet Protocol, Src Addr: 217.128.147.4
                   Dst Addr: 195.25.12.28
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00
                            (DSCP 0x00: Default; ECN: 0)
      0000 00..=Differentiated Services
                               Codepoint:Default (0)
      .... ..0.=ECN-Capable Transport (ECT): 0
      .... ...0=ECN-CE: 0
    Total Length: 84
    Identification: 0x0000
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
    Header checksum: 0xfeee (correct)
    Source: 217.128.147.4
    Destination: 195.25.12.28
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf320 (correct)
    Identifier: 0x0e10
    Sequence number: 00:00
    Data (56 bytes)


0000  7c a6 07 3c 87 e9 00 00 08 09 0a 0b 0c 0d 0e 0f
```

```
    No link information available
**
**
** Au niveau ppp, ce qu'il se passe en dessous
** interprété, le renifleur ne sait pas le faire,
** il se croit sur un "vrai" lien PPP
** parce qu'il écoute sur une interface PPP.
** Mais en regardant au niveau Ethernet
** Nous trouvons les informations relatives
** au protocole PPPoE
**
Internet Protocol, Src Addr: 217.128.147.4
                   Dst Addr: 195.25.12.28
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00
                            (DSCP 0x00: Default; ECN:0)
      0000 00..=Differentiated Services
                               Codepoint: Default (0)
      .... ..0.=ECN-Capable Transport (ECT): 0
      .... ...0=ECN-CE: 0
    Total Length: 84
    Identification: 0x0000
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
    Header checksum: 0xfeee (correct)
    Source: 217.128.147.4 (217.128.147.4)
    Destination: 195.25.12.28 (195.25.12.28)
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf320 (correct)
    Identifier: 0x0e10
    Sequence number: 00:00
    Data (56 bytes)


0000  7c a6 07 3c 87 e9 00 00 08 09 0a 0b 0c 0d 0e 0f
```

Sur Ethernet (Eth1)

```
0010  10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
0020  20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
0030  30 31 32 33 34 35 36 37

Frame 5 (106 on wire, 106 captured)
    Arrival Time: Nov 30, 2001 16:32:12.122676
    Time delta from previous packet: 0.062395 seconds
    Time relative to first packet: 8.699354 seconds
    Frame Number: 5
    Packet Length: 106 bytes
    Capture Length: 106 bytes
Ethernet II
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 218c
    Payload Length: 86
Point-to-Point Protocol
    Protocol: IP (0x0021)
Internet Protocol, Src Addr: 195.25.12.28
                   Dst Addr: 217.128.147.4
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00
                            (DSCP 0x00: Default; ECN: 0)
      0000 00..=Differentiated Services
                               Codepoint:Default (0)
      .... ..0.=ECN-Capable Transport (ECT): 0
      .... ...0=ECN-CE: 0
    Total Length: 84
    Identification: 0xf960
    Flags: 0x00
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 248
    Protocol: ICMP (0x01)
```

Sur PPPoE (ppp0)

```
0010  10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
0020  20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f
0030  30 31 32 33 34 35 36 37

Frame 2 (84 on wire, 84 captured)
    Arrival Time: Nov 30, 2001 16:32:12.122954
    Time delta from previous packet: 0.062962 seconds
    Time relative to first packet: 0.062962 seconds
    Frame Number: 2
    Packet Length: 84 bytes
    Capture Length: 84 bytes
Raw packet data
    No link information available
**
**
**
**
** Et pour la réponse
** c'est la même chose
**
**
**
**
Internet Protocol, Src Addr: 195.25.12.28
                   Dst Addr: 217.128.147.4
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00
                            (DSCP 0x00: Default; ECN: 0)
      0000 00..=Differentiated Services
                               Codepoint:Default (0)
      .... ..0.=ECN-Capable Transport (ECT): 0
      .... ...0=ECN-CE: 0
    Total Length: 84
    Identification: 0xf960
    Flags: 0x00
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 248
    Protocol: ICMP (0x01)
```

Sur Ethernet (Eth1)	Sur PPPoE (ppp0)
Header checksum: 0x8d8d (correct) Source: 195.25.12.28 (195.25.12.28) Destination: 217.128.147.4 (217.128.147.4) Internet Control Message Protocol Type: 0 (Echo (ping) reply) Code: 0 Checksum: 0xfb20 (correct) Identifier: 0x0e10 Sequence number: 00:00 Data (56 bytes) 0000 7c a6 07 3c 87 e9 00 00 08 09 0a 0b 0c 0d 0e 0f 0010 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 0020 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 0030 30 31 32 33 34 35 36 37	Header checksum: 0x8d8d (correct) Source: 195.25.12.28 Destination: 217.128.147.4 Internet Control Message Protocol Type: 0 (Echo (ping) reply) Code: 0 Checksum: 0xfb20 (correct) Identifier: 0x0e10 Sequence number: 00:00 Data (56 bytes) 0000 7c a6 07 3c 87 e9 00 00 08 09 0a 0b 0c 0d 0e 0f 0010 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 0020 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 0030 30 31 32 33 34 35 36 37

Comme il est facile de le constater, on a bien de l'IP qui est transporté au dessus de PPP, lui même au dessus d'Ethernet. Si ce n'est la couche supplémentaire introduite par ce protocole, tout reste identique à ce que l'on observerait sur un réseau Ethernet "classique".

Les détails

Mise en confiance

Ce qui va suivre peut paraître quelque peu "indigeste". Il n'est donc peut-être pas inutile de donner quelques points de repères avant d'entamer cette descente aux enfers.

Nous savons maintenant que le but ultime est d'exploiter les possibilités de PPP sur un réseau de nature Ethernet, parce que PPP offre quelques facilités aux fournisseurs de services, comme l'identification nominative des clients, principalement. C'est nécessaire lorsqu'il y a plusieurs fournisseurs d'accès qui cohabitent sur la même structure.

Dans la page précédente, nous avons vu qu'une fois la connexion PPP établie, IP est transporté par PPP (oE), lui-même transporté par Ethernet.

Nous devons donc nous attendre, lors de l'établissement de cette connexion, à observer comment PPPoE va s'y prendre pour mettre en place le lien PPP entre notre machine et celle de notre fournisseur de services. Nous nous arrêterons lorsque PPP sera établi, ça suffira. Le reste concernerait le protocole PPP lui-même, ce qui n'est pas l'objectif de cet exposé.

- Nous sommes sur un réseau Ethernet, donc en architecture de réseau, plusieurs hôtes sont présents sur ce réseau et parmi eux, il y a celui avec lequel il faut mettre en place le lien PPP. Il va donc falloir identifier cet interlocuteur sur ce réseau. Le seul moyen connu au niveau Ethernet, c'est un "broadcast ARP" (Diffusion sur toutes les adresses MAC présentes). Une au moins des machines du fournisseur d'accès devrait répondre.
  Une fois les deux interlocuteurs mutuellement reconnus, il n'y aura plus de broadcast ARP. Comme nous allons le voir, la reconnaissance mutuelle va aboutir à l'octroi d'un identifiant de session qui restera valide tout le temps de la session.
- PPP, au moyen du sous-protocole LCP (Link Control Protocol, protocole spécialisé dans la négociation et la maintenance de la connexion PPP), va identifier le client (Nom d'utilisateur et mot de passe).
- Si cette identification réussit, LCP va fournir au client les paramètres nécessaires pour le bon fonctionnement d'IP :
  - Adresse IP du client.
  - Serveur DNS pour la résolution des noms.
  - Passerelle par défaut. (Ici, cette passerelle est symbolique, puisque sur la connexion PPP, il n'y a que deux protagonistes : Vous et l'équipement de votre FAI à l'autre bout. C'est forcément lui la passerelle).
- Si l'identification échoue, la ligne sera "raccrochée" (par analogie avec un modem RTC). Il faudra donc reprendre la connexion à son début.

Voici donc en quelques mots, ce que nous devrions vérifier dans la suite immédiate. Accrochez-vous, on y va.

RFC...

Les "Request For Comment" sont une très grande chose :

1. Comme leur nom ne l'indique pas du tout, elles sont publiées une fois que le texte est finalisé et qu'il n'y a plus de commentaires à faire à son sujet.
2. Elles décrivent généralement dans le détail les divers protocoles utilisés sur l'Internet (dont PPPoE, bien entendu) et toute personne mettant en oeuvre un protocole de l'Internet se doit de le faire en respectant les RFCs qui le décrivent, c'est l'assurance que ce protocole sera utilisable par tous.
3. Elles sont initialement rédigées en Anglais, par des spécialistes au langage particulièrement obscur.
4. A cause de toutes les propriétés citées plus haut, elles servent d'argument "massue" à ceux qui veulent à tout prix montrer qu'ils sont les plus compétents et qu'ils planent bien au dessus des foules ignares (Une réponse communément trouvée sur les newsgroups : "Va d'abord lire les RFC...").
5. A cause du point 2 (le seul positif), elles sont tout de même d'une utilité inestimable.

Par chance pour nous, plusieurs personnes se sont attelées à l'ingrate tâche de la traduction de ces RFCs. Toutes ne le sont pas encore, mais la RFC 2516, celle qui décrit le protocole PPPoE, est traduite[6].

Lisez cette RFC, vous constaterez combien le point 3, même affranchi de la langue Anglaise, reste vérifié. Lisez la quand même si vous voulez vraiment connaître le détail de ce protocole.

Pour vous aider un peu dans cette lecture...

Voici la manipulation proposée :

- Une machine Linux Mandrake 8.1 est connectée à une liaison Netissimo (France Télécom) via un modem Ethernet SpeedTouch Home (Alcatel).
- Le client PPPoE utilisé est rp-pppoe.
- Nous ouvrons une session PPPoE, un renifleur est à l'écoute, qui récupère tout ce qu'il se passe au niveau Ethernet.
- Nous comparons ce que nous voyons avec ce qui est dit dans les RFC.

6 RFC 2516 en français : http://abcdrfc.free.fr/rfc-vf/rfc2516.html

Ce que disent les Textes

L'étape de découverte

l'étape de découverte s'effectue en quatre étapes. Quand elle s'achève, chaque vis à vis connaît le PPPOE SESSION_ID ainsi que les adresses Ethernet ; cela suffit pour définir une session PPPOE. Les étapes sont :

- *Emission d'un paquet broadcast d'initiation par l'hôte ;*
- *Emission de paquets d'offres par un concentrateur d'accès ou plus ;*
- *Emission d'un paquet de demande de session unicast par l'hôte ;*
- *Et émission d'un paquet de confirmation par le concentrateur d'accès.*

Après avoir envoyé le paquet de confirmation et dès que l'hôte le reçoit, la connexion passe alors à l'étape suivante : la session PPP.

Toutes les trames de découvertes Ethernet ont le champ ETHER_TYPE à 0x8863.

Ce que nous pouvons observer

Etablissement de PPPoE

Dans un premier temps, juste le résumé des trames qui passent :

```
No.  Source             Destination        Protocol   Info
 4   3Com_50:f0:df      ff:ff:ff:ff:ff:ff  PPPoED     Active Discovery Initiation (PADI)
 5   Redback_00:4f:7d   3Com_50:f0:df      PPPoED     Active Discovery Offer (PADO)
 6   3Com_50:f0:df      Redback_00:4f:7d   PPPoED     Active Discovery Request (PADR)
 7   Redback_00:4f:7d   3Com_50:f0:df      PPPoED     Active Discovery Session-confirmation (PADS)
 8   Redback_00:4f:7d   3Com_50:f0:df      PPP LCP    PPP LCP Configuration Request
```

Et voici, exprimée dans toute sa beauté, la magie des systèmes bien normalisés : ça va se passer exactement comme c'est dit dans les textes.

Exactement ? Voyons ça de plus près...

Le paquet PADI (PPPoE Active Discovery Initiation) :

Les hôtes envoient en broadcast un paquet PADI. Le champ CODE est mis à 0x09 et le champ SESSION_ID à 0x0000.

Le paquet PADI doit contenir un TAG de type Service-Name, indiquant le service que l'hôte demande ainsi que d'autres numéros correspondant à d'autres types de TAG. Un paquet PADI entier (incluant l'en-tête PPPoE) ne doit pas dépasser 1484 octets afin de laisser la place suffisante pour qu'un agent relais puissent ajouter un TAG Relay-Session-Id.

> ***Note:*** *A l'usage, j'ai pu constater que les rapports d'analyse de trames affichés "en français" peuvent induire en erreur. Ces rapports sont générés par le renifleur (sniffer), parce qu'il connaît par coeur le format des trames qu'il capture et qu'il les interprète de façon plus "lisible". Dans la pratique, les données capturées ne sont rien de plus que la suite d'octets, ici surlignés.Dans la trame qui suit, il n'y en a que 32, qui génèrent 24 lignes "d'explications".*

```
0000   ff ff ff ff ff ff 00 60 8c 50 f0 df 88 63 11 09    Ce sont les information effectivement
capturées
0010   00 00 00 0c 01 01 00 00 01 03 00 04 3d 53 00 00
```

```
.............................................

Frame 4 (32 on wire, 32 captured)                          Ceci n'est que du calcul réalisé par le
sniffer
    Arrival Time: Dec  3, 2001 15:12:09.426679           ces informations ne correspondent à aucune
donnée
    Time delta from previous packet: 10.824602 seconds   écrite dans la trame
    Time relative to first packet: 11.398184 seconds
    Frame Number: 4
    Packet Length: 32 bytes
    Capture Length: 32 bytes
.............................................
Ethernet II                                              Les informations capturées commencent ici
    Destination: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)   ff ff ff ff ff ff
    *** La destination est bien un "broadcast"
        sur les adresses MAC
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)            00 60 8c 50 f0 df
    *** La source est l'adresse MAC
        de l'interface Ethernet
    *** connectée au modem ADSL
    Type: PPPoE Discovery (0x8863)                       88 63
    *** ETHER_TYPE est bien à 8863H
PPP-over-Ethernet Discovery
    Version: 1
    Type: 1                                              11
    Code: Active Discovery Initiation (PADI)             09
    *** C'est bien une trame PADI Le champ "code" est à 9
    Session ID: 0000                                     00 00
    *** Le champ "Session_ID" est bien à 0
    Payload Length: 12                                   00 0c
PPPoE Tags
    Tag: Service-Name                                    01 01 00 00
    *** Le Tag "Service-Name, comme indiqué
    Tag: Host-Uniq                                       01 03 00 04
    *** Et un autre Tag: Host-Uniq                       3d 53 00 00
      Binary Data: (4 bytes)
```

Remarquez la similitude avec DHCP discovery[7]. Le client qui se "réveille" envoie un broadcast ARP pour trouver un interlocuteur qui devra lui indiquer ses paramètres de configuration.

Le Tag "Host-Uniq" est décrit dans l'annexe A:

> ***0x0103 Host-Uniq***
>
> *Ce Tag est utilisé par un hôte pour associer de façon unique la réponse d'un concentrateur d'accès (PADO ou PADS) à la requête d'un hôte particulier (PADI ou PADR). Sa valeur est une donnée binaire de n'importe quelle valeur et de n'importe quelle longueur, choisies par l'hôte. Cette valeur n'est pas interprétée par le concentrateur d'accès.*
>
> *Un hôte PEUT inclure un Tag "Host-Uniq" dans un paquet PADI ou PADR. Si le concentrateur d'accès reçoit ce Tag, il DOIT inclure ce Tag sans le modifier dans la réponse PADO ou PADS associée.*

Ce Tag (0x0103) est suivi du nombre d'octets qu'il contient (0x0004) et des octets de données (0x3d530000). Nous devrions donc théoriquement retrouver ce Tag dans son intégralité dans la réponse PADO qui suit.

Il n'y a pas ici d'agent de relais. Pour l'instant, tout est donc bien conforme.

***Note**: Le "Payload" est assez difficile à bien traduire, peut-être par "Données utiles" (mais elles sont toutes utiles). Comme en mot-à-mot, ça donne : "la charge qui paye", nous dirons "charge utile". En s'appuyant sur cette analyse de trames, on constate que le "payload length" n'est autre que le nombre d'octets qui suivent, ce qui sera confirmé dans la*

7 DHCP : http://christian.caleca.free.fr/dhcp/protocole_dhcp.htm

suite.
Ce payload contient des "tags", un peu plus facile à traduire...

> ***Le paquet PADO (PPPoE Active Discovery Offer)***
>
> *Quand le concentrateur d'accès reçoit un PADI qu'il peut servir, il répond en envoyant un paquet PADO. l'adresse de destination est l'adresse unicast de l'hôte envoyé dans le PADI. Le champ CODE est fixé à 0x07 et le champ SESSION_ID à 0x0000.*
>
> *Le paquet PADO doit contenir un TAG AC-Name : c'est le nom du concentrateur d'accès, un TAG Service-Name identique à celui contenu dans le PADI. Les autres numéros correspondent aux autres services qui peuvent être offerts par le concentrateur d'accès. Si le concentrateur d'accès ne peut pas servir le PADI alors celui-ci ne répond pas avec un PADO.*

Le client qui a démarré sa connexion PPPoE vient d'essayer de découvrir un interlocuteur, le ou les interlocuteurs présents vont maintenant lui répondre.

```
0000   00 60 8c 50 f0 df 00 02 3b 00 4f 7d 88 63 11 07
0010   00 00 00 2b 01 01 00 00 01 03 00 04 3d 53 00 00
0020   01 02 00 17 36 32 30 33 32 30 33 30 31 30 38 33
0030   37 36 2d 42 53 4d 41 52 31 30 32 01 01 00 00

Frame 5 (63 on wire, 63 captured)
    Arrival Time: Dec  3, 2001 15:12:09.479615
    Time delta from previous packet: 0.052936 seconds
    Time relative to first packet: 11.451120 seconds
    Frame Number: 5
    Packet Length: 63 bytes
    Capture Length: 63 bytes
Ethernet II
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)     00 60 8c 50 f0 df
    *** La destination est ici le client
    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)       00 02 3b 00 4f 7d
    *** Et la source, l'équipement
        du gestionnaire du réseau
    Type: PPPoE Discovery (0x8863)                     88 63
    *** C'est toujours un type "Discovery"
PPP-over-Ethernet Discovery
    Version:1                                          11
    Type: 1
    Code: Active Discovery Offer (PADO)                07
    *** Mais ici, c'est un "Offer"
    Session ID: 0000                                   00 00
    Payload Length: 43  00 2b
PPPoE Tags
    Tag: Service-Name                                  01 01 00 00
    *** Nous retrouvons, comme prévu,
        le Tag "Host-Uniq"
    Tag: Host-Uniq                                     01 03 00 04
      Binary Data: (4 bytes)                           3d 53 00 00
    *** Le Tag "AC_NAME"...
    Tag: AC-Name                                       01 02 00 17
    *** Et sa valeur (nom du concentrateur d'accès)
      String Data: 62032030108376-BSMAR102             36 32 30 33 32 30 33 30 31 30 38 33
                                                       37 36 2d 42 53 4d 41 52 31 30 32
    Tag: Service-Name                                  01 01 00 00
```

Ca devient monotone, il n'y a aucune surprise... Tant pis pour le "suspense", il n'y aura pas d'autres réponses PADO. Nous allons donc maintenant retrouver l'hôte qui envoie un paquet PADR (Session-Request).

Le paquet PADR (PPPoE Active Discovery Request)

Puisque le PADI a été envoyé en broadcast l'hôte peut recevoir plusieurs PADO. L'hôte examine les paquets PADO reçus et en choisit un. Le choix peut être basé sur le nom du concentrateur d'accès ou sur les services offerts. L'hôte envoie alors un paquet PADR au concentrateur d'accès sélectionné. Le champ DESTINATION_ADDR est l'adresse Ethernet unicast du concentrateur d'accès qui a envoyé par le PADO. Le champ CODE est 0x19 et le champ SESSION_ID est à la valeur 0x0000.

Le paquet PADR doit contenir exactement un TAG_TYPE contenant le nom du service que l'hôte demande ainsi que d'autres numéros d'autres types de TAG.

```
0000  00 02 3b 00 4f 7d 00 60 8c 50 f0 df 88 63 11 19
0010  00 00 00 0c 01 01 00 00 01 03 00 04 3d 53 00 00

Frame 6 (32 on wire, 32 captured)
    Arrival Time: Dec  3, 2001 15:12:09.480206
    Time delta from previous packet: 0.000591 seconds
    Time relative to first packet: 11.451711 seconds
    Frame Number: 6
    Packet Length: 32 bytes
    Capture Length: 32 bytes
Ethernet II
    *** Vous avez compris maintenant le niveau Ethernet...
    Destination: 00:02:3b:00:4f:7d (Redback_00:4f:7d)          00 02 3b 00 4f 7d
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)                  00 60 8c 50 f0 df
    Type: PPPoE Discovery (0x8863)                             88 63
    *** Et je vous laisse faire la suite tout seuls...
PPP-over-Ethernet Discovery
    Version: 1
    Type: 1                                                    11
    Code: Active Discovery Request (PADR)                      19
    Session ID: 0000                                           00 00
    Payload Length: 12                                         00 0c
PPPoE Tags
    Tag: Service-Name                                          01 01 00 00
    Tag: Host-Uniq                                             01 03 00 04
      Binary Data: (4 bytes)                                   3d 53 00 00
```

Il n'y a pas de grosses différences avec le paquet PADI, si ce n'est que l'adresse du destinataire n'est plus une adresse de broadcast, mais celle du concentrateur d'accès, puisque maintenant, on la connaît.

Finalement, le Concentrateur d'accès va confirmer cette connexion :

Le paquet PADS(PPPoE Active Discovery Session-confirmation)

Quand le Concentrateur d'Accès reçoit un paquet PADR, il se prépare à commencer une session PPP. Il produit un SESSION_ID unique pour la session PPPOE et répond à l'hôte avec un paquet PADS. Le champ DESTINATION_ADDR est l'adresse Ethernet unicast de l'hôte qui a envoyé le PADR. Le champ CODE est mis à 0x65 et le SESSION_ID DOIT être mis à la valeur unique produite pour cette session PPPOE.

Le paquet PADS contient exactement un TAG_TYPE contenant le nom du service sous lequel le concentrateur d'accès a accepté la session PPPoE et d'autres numéros pour d'autres types de TAG.

Si le concentrateur d'accès n'accepte pas le service proposé dans le PADR, il doit répondre avec des PADS contenant TAG _TYPE Service-Name-Error (et d'autres numéros d'autres TAG). Dans ce cas le SESSION_ID DOIT être à la valeur 0x0000.

```
0000   00 60 8c 50 f0 df 00 02 3b 00 4f 7d 88 63 11 65
0010   02 f4 00 27 01 01 00 00 01 03 00 04 3d 53 00 00
0020   01 02 00 17 36 32 30 33 32 30 33 30 31 30 38 33
0030   37 36 2d 42 53 4d 41 52 31 30 32 3d

Frame 7 (60 on wire, 60 captured)
    Arrival Time: Dec  3, 2001 15:12:09.547915
    Time delta from previous packet: 0.067709 seconds
    Time relative to first packet: 11.519420 seconds
    Frame Number: 7
    Packet Length: 60 bytes
    Capture Length: 60 bytes
Ethernet II
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)        00 60 8c 50 f0 df
    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)          00 02 3b 00 4f 7d
    Type: PPPoE Discovery (0x8863)                        88 63
PPP-over-Ethernet Discovery
    Version: 1
    Type: 1                                               11
    Code: Active Discovery Session-confirmation (PADS)    65
    *** Nous récupérons ici la Session-ID
    Session ID: 02f4                                      02 f4
    Payload Length: 43                                    00 2b
PPPoE Tags
    Tag: Service-Name                                     01 01 00 00
    Tag: Host-Uniq                                        01 03 00 04
      Binary Data: (4 bytes)                              3d 53 00 00
    Tag: AC-Name                                          01 02 00 17
      String Data: 62032030108376-BSMAR102                36 32 30 33 32 30 33 30 31 30 38 33
                                                          37 36 2d 42 53 4d 41 52 31 30 32
```

Il n'y a pas eu de problèmes, la session est acceptée par les deux partenaires et elle aura l'identifiant 0x02f4. Nous retrouverons systématiquement cet identifiant dans tous les paquets qui suivront.

C'est fini pour l'établissement de la session PPPoE. Comme vous avez pu le remarquer, c'est assez simple et il n'y a pas grand chose de fait. Tout de même, faisons un petit bilan :

- L'hôte client a cherché et trouvé un Concentrateur d'accès.
- Le Concentrateur d'accès a délivré à l'hôte client :
  - Son adresse MAC (ici : 00:02:3b:00:4f:7d)
  - Un numéro de session PPPoE (ici : 0x02f4)

Etablissement de PPP

Tout ceci est très bien, mais nous n'avons pas d'adresse IP, ni de passerelle, ni de DNS... Autant de choses nécessaires pour faire fonctionner correctement TCP/IP, sans oublier que, pour l'instant, nous ne sommes toujours pas authentifiés.

Le reste va maintenant être négocié par le protocole PPP, de la même manière qu'avec une connexion "classique" par modem RTC.

PPP est encore une autre affaire, qui dépasse le cadre de ce chapitre. Nous n'allons donc pas étudier par le détail les trames qui suivent. Nous allons tout de même regarder comment les informations qui nous manquent pour l'instant vont être récupérées. Si vous désirez absolument approfondir PPP, vous avez les RFC 1661 traduites en français ici[8].

Voici le sommaire des trames qui nous intéressent :

```
 8  Redback_00:4f:7d   3Com_50:f0:df      PPP LCP    PPP LCP Configuration Request
 9  3Com_50:f0:df      Redback_00:4f:7d   PPP LCP    PPP LCP Configuration Request
10  3Com_50:f0:df      Redback_00:4f:7d   PPP LCP    PPP LCP Configuration Ack
11  Redback_00:4f:7d   3Com_50:f0:df      PPP LCP    PPP LCP Configuration Ack
12  3Com_50:f0:df      Redback_00:4f:7d   PPP LCP    PPP LCP Echo Request
13  Redback_00:4f:7d   3Com_50:f0:df      0xc223     PPP Cryptographic Handshake Auth. Protocol
(0xc223)
14  3Com_50:f0:df      Redback_00:4f:7d   0xc223     PPP Cryptographic Handshake Auth. Protocol
(0xc223)
15  Redback_00:4f:7d   3Com_50:f0:df      PPP LCP    PPP LCP Echo Reply
16  Redback_00:4f:7d   3Com_50:f0:df      0xc223     PPP Cryptographic Handshake Auth. Protocol
(0xc223)
17  Redback_00:4f:7d   3Com_50:f0:df      PPP LCP    PPP LCP Echo Request
18  Redback_00:4f:7d   3Com_50:f0:df      PPP IPCP   PPP IPCP Configuration Request
19  3Com_50:f0:df      Redback_00:4f:7d   PPP IPCP   PPP IPCP Configuration Request
20  3Com_50:f0:df      Redback_00:4f:7d   PPP LCP    PPP LCP Echo Reply
21  3Com_50:f0:df      Redback_00:4f:7d   PPP IPCP   PPP IPCP Configuration Ack
22  Redback_00:4f:7d   3Com_50:f0:df      PPP IPCP   PPP IPCP Configuration Nak
23  3Com_50:f0:df      Redback_00:4f:7d   PPP IPCP   PPP IPCP Configuration Request
24  Redback_00:4f:7d   3Com_50:f0:df      PPP IPCP   PPP IPCP Configuration Ack
25  Redback_00:4f:7d   3Com_50:f0:df      IP         Bogus IP header length (0, must be at least 20)
26  Redback_00:4f:7d   3Com_50:f0:df      PPP LCP    PPP LCP Echo Request
27  3Com_50:f0:df      Redback_00:4f:7d   PPP LCP    PPP LCP Echo Reply
```

Nous n'allons pas les détailler, mais juste en extraire les points les plus importants. Déjà, nous allons sauter les LCP Echo request et reply, qui ne présentent pas un intérêt fondamental pour ce qui nous intéresse, si ce n'est qu'ils sont recommandés par les RFC.

Le protocole LCP (Link Control Protocol) est transporté par PPP. Sa fonction, comme son nom l'indique, est de maintenir le bon état du lien PPP. En particulier, c'est lui qui va permettre de négocier au départ la configuration IP du client, ce que nous allons voir tout de suite.

Demande de configuration de la part du concentrateur d'accès

```
Frame 8 (60 on wire, 60 captured)
    ...
Ethernet II                                          Ethernet
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)   La cible est le client

    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)     La source est le Concentrateur d'accès
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session                            PPPoE
    ...
    Code: Session Data
    Session ID: 02f4                                 Le Session-ID qui a été négocié précédemment
    Payload Length: 21
Point-to-Point Protocol                              PPP
```

8 RFC 1661 en franais : http://abcdrfc.free.fr/rfc-vf/rfc1661.html

```
    Protocol: Link Control Protocol (0xc021)
PPP Link Control Protocol                          LCP
    Code: Configuration Request (0x01)
    Identifier: 0x6d
    Length: 19
    Options: (15 bytes)
        MRU: 1492                                  Très important!!!, nous y reviendrons.
L'identification va être cryptée
        Authentication protocol: 5 bytes           (Cryptographic Handshake Auth. Protocol)
            Authentication protocol: CHAP (0xc223) Le "Nombre Magique" est sans intérêt pour nous
            Data (1 byte)(voir RFC 1661)
        Magic number: 0x2889d071
```

MRU: Maximum Receive Unit, taille maximale en octets d'un paquet acceptable en réception. C'est la valeur qui devra être adoptée pour le MTU (Maximum Transfert Unit) de l'interlocuteur.

Demande de configuration de la part du client.

```
Frame 9 (36 on wire, 36 captured)                  Le client n'a pas grand chose à réclamer
    ...                                            pour la configuration...
Ethernet II
    Destination: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 16
Point-to-Point Protocol
    Protocol: Link Control Protocol (0xc021)
PPP Link Control Protocol
    Code: Configuration Request (0x01)
    Identifier: 0x01
    Length: 14
    Options: (10 bytes)
        MRU: 1492                                  Le MRU, également à 1492.
        Magic number: 0x694c9902
```

Mais pourquoi 1492 ? Une trame Ethernet ne doit pas dépasser 1500 Octets. Comme l'en-tête de PPPoE fait 6 octets et que le PPP_ID est de 2 octets, il ne reste que 1492 octets pour le PPP_MTU.

Acquittement du client

```
Frame 10 (41 on wire, 41 captured)                 Le client est d'accord pour :
    ...
Ethernet II
    Destination: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 21
Point-to-Point Protocol
    Protocol: Link Control Protocol (0xc021)
PPP Link Control Protocol
    Code: Configuration Ack (0x02)
    Identifier: 0x6d
    Length: 19
    Options: (15 bytes)
        MRU: 1492                                  Le MRU à 1492
        Authentication protocol: 5 bytes
            Authentication protocol: CHAP (0xc223) le protocole CHAP pour l'authentification
            Data (1 byte)
        Magic number: 0x2889d071                   Et le numéro magique
```

Acquittement du concentrateur d'accès

```
Frame 11 (60 on wire, 60 captured)                  Le concentrateur d'accès est d'accord pour:
    ...
Ethernet II
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 16
Point-to-Point Protocol
    Protocol: Link Control Protocol (0xc021)
PPP Link Control Protocol
    Code: Configuration Ack (0x02)
    Identifier: 0x01
    Length: 14
    Options: (10 bytes)
        MRU: 1492                                   Le MRU à 1492
        Magic number: 0x694c9902                    Et le numéro magique.
```

Authentification du Concentrateur

```
Frame 13 (60 on wire, 60 captured)                                  Le concentrateur envoie au
                                                                    client un paquet
    ...                                                             un paquet de données...
Ethernet II
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 31
Point-to-Point Protocol
    Protocol: Cryptographic Handshake Auth. Protocol (0xc223)
Data (38 bytes)

0000  00 60 8c 50 f0 df 00 02 3b 00 4f 7d 88 64 11 00  .`.P....;.O}.d..   Le détail de ce paquet
                                                                    pourrait être
0010  02 f4 00 1f c2 23 01 01 00 1d 10 75 51 f4 58 40  .....#.....uQ.X@   décrit en étudiant le
                                                                    protocole CHAP...
0020  38 9b 08 c2 8f 76 9f 8e 89 81 3c 42 53 4d 41 52  8....v....<BSMAR   Notez que le nom du
                                                                    concentrateur
0030  31 30 32 66 00 50 35 d1 cc 8f 00 00              102f.P5.....       apparaît en clair
```

Authentification du client

```
Frame 14 (58 on wire, 58 captured)                                  Le client envoie au
                                                                    concentrateur
    ...                                                             un paquet de données...
Ethernet II
    Destination: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 38
Point-to-Point Protocol
    Protocol: Cryptographic Handshake Auth. Protocol (0xc223)
Data (36 bytes)

0000  00 02 3b 00 4f 7d 00 60 8c 50 f0 df 88 64 11 00   ..;.O}.`.P...d.. Le détail de ce paquet
```

```
                                                                    pourrait être
0010  02 f4 00 26 c2 23 02 01 00 24 10 38 af 00 96 c1   ...&.#...$.8.... décrit en étudiant le
                                                                    protocole CHAP...
0020  b0 95 b2 b2 ee 6f be bb d4 cd 5e 66 74 69 2f xx   .....o....^fti/x Notez que le nom du client
                                                                    apparaît
0030  xx xx xx xx xx xx 40 66 74 69                     xxxxxx@fti       également en clair
                                                                    (les x, c'est de ma part).
```

Le verdict...

```
Frame 16 (61 on wire, 61 captured)
    ...
Ethernet II
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 41
Point-to-Point Protocol
    Protocol: Cryptographic Handshake Auth. Protocol (0xc223)
Data (39 bytes)

0000  00 60 8c 50 f0 df 00 02 3b 00 4f 7d 88 64 11 00   .`.P....;.O}.d.. On ne sait pas exactement
                                                                    comment, mais...
0010  02 f4 00 29 c2 23 03 01 00 27 43 48 41 50 20 61   ...).#...'CHAP a L'authentification a
                                                                    réussi.
0020  75 74 68 65 6e 74 69 63 61 74 69 6f 6e 20 73 75   uthentication su
0030  63 63 65 73 73 2c 20 75 6e 69 74 20 39            ccess, unit 9
```

Requête de configuration du concentrateur

```
Frame 18 (60 on wire, 60 captured)                  A vrai dire
    ...                                             Ce n'est pas exactement une requête

Ethernet II
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)  Le concentrateur n'a rien à demander

    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 12
Point-to-Point Protocol
    Protocol: IP Control Protocol (0x8021)
PPP IP Control Protocol
    Code: Configuration Request (0x01)              il annonce juste son IP
    Identifier: 0x6e
    Length: 10
    Options: (6 bytes)
        IP address: 217.128.147.1
```

Requête de configuration du client

```
Frame 19 (44 on wire, 44 captured)                      Ici, c'est plus amusant.
    ...                                                 Le client propose une configuration
Ethernet II
    Destination: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
```

```
    Session ID: 02f4
    Payload Length: 24
Point-to-Point Protocol
    Protocol: IP Control Protocol (0x8021)
PPP IP Control Protocol
    Code: Configuration Request (0x01)
    Identifier: 0x01
    Length: 22
    Options: (18 bytes)
        IP address: 0.0.0.0                          Comme il se réveille juste, il propose
        Primary DNS server IP address: 0.0.0.0       une configuration plutôt farfelue...
        Secondary DNS server IP address: 0.0.0.0
```

Acquittement de la configuration du concentrateur

```
Frame 21 (32 on wire, 32 captured)
    ...
Ethernet II
    Destination: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 12
Point-to-Point Protocol
    Protocol: IP Control Protocol (0x8021)
PPP IP Control Protocol
    Code: Configuration Ack (0x02)                   Le client est d'accord...
    Identifier: 0x6e
    Length: 10
    Options: (6 bytes)                               Pour l'adresse IP du concentrateur
        IP address: 217.128.147.1                    (qui deviendra sa passerelle par défaut)
```

Refus de la configuration du client

```
Frame 22 (60 on wire, 60 captured)                   On s'en souvient, le client avait
    ...                                              proposé une configuration stupide.
Ethernet II
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)   En fait, elle ne l'était pas.
    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)     Ne connaissant pas sa configuration,
    Type: PPPoE Session (0x8864)                     le client indique des adresses
PPP-over-Ethernet Session                            toutes à zéro.
    Version: 1
    Type: 1                                          C'est dans le but que le concentrateur
    Code: Session Data                               lui propose à la place des adresses
    Session ID: 02f4                                 valides
    Payload Length: 24
Point-to-Point Protocol
    Protocol: IP Control Protocol (0x8021)
PPP IP Control Protocol
    Code: Configuration Nak (0x03)
    Identifier: 0x01
    Length: 22
    Options: (18 bytes)
        IP address: 217.128.147.4                    Son IP
        Primary DNS server IP address: 193.252.19.3  Et le (ici les) DNS.

        Secondary DNS server IP address: 193.252.19.4
```

Re requête de configuration du client

```
Frame 23 (44 on wire, 44 captured)                   Un protocole, c'est parfois bête,
    ...                                              mais toujours discipliné.
Ethernet II
    Destination: 00:02:3b:00:4f:7d (Redback_00:4f:7d)
    Source: 00:60:8c:50:f0:df (3Com_50:f0:df)
    Type: PPPoE Session (0x8864)
```

```
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 24
Point-to-Point Protocol
    Protocol: IP Control Protocol (0x8021)
PPP IP Control Protocol
    Code: Configuration Request (0x01)
    Identifier: 0x02
    Length: 22                                  Le client refait une requête de configuration
    Options: (18 bytes)                         cette fois-ci avec les adresses qui lui ont
été
        IP address: 217.128.147.4               "suggérées" par le concentrateur...
        Primary DNS server IP address: 193.252.19.3
        Secondary DNS server IP address: 193.252.19.4
```

Acquittement de la configuration du client

```
Frame 24 (60 on wire, 60 captured)              Le concentrateur, sous peine de passer
    ...                                         pour un fou,
Ethernet II                                     ne pourra pas faire autrement (en principe)
    Destination: 00:60:8c:50:f0:df (3Com_50:f0:df)  que d'accepter cette configuration
    Source: 00:02:3b:00:4f:7d (Redback_00:4f:7d)    qu'il a proposé lui-même.
    Type: PPPoE Session (0x8864)
PPP-over-Ethernet Session
    Version: 1
    Type: 1
    Code: Session Data
    Session ID: 02f4
    Payload Length: 24
Point-to-Point Protocol
    Protocol: IP Control Protocol (0x8021)
PPP IP Control Protocol
    Code: Configuration Ack (0x02)              Voilà qui est fait.
    Identifier: 0x02
    Length: 22
    Options: (18 bytes)
        IP address: 217.128.147.4
        Primary DNS server IP address: 193.252.19.3
        Secondary DNS server IP address: 193.252.19.4
```

Toute cette séquence n'aura duré que 3 secondes.

- Établissement d'une session PPPoE entre le client et le concentrateur, avec établissement d'un identifiant de session.
- Authentification du client de la part du concentrateur par LCP (sur PPP).
- Obtention d'une configuration valide pour le client (Adresse IP, Adresses de DNS et Passerelle, la passerelle étant bien entendu le concentrateur lui-même).

Conclusions

J'espère que cet exposé vous aura éclairé sur le fonctionnement de ce protocole qui tend à devenir universel sur les connexions "haut débit". Les points principaux à retenir me semble être les suivants :

- Établissement d'une connexion PPP au dessus d'Ethernet avec obtention d'un identifiant de session.
- Identification du client avec LCP, protocole de gestion et de maintenance d'une connexion PPP.

- Obtention des paramètres de connexion IP par ce même protocole LCP.
- Transport des datagrammes IP par PPP, lui même transporté par Ethernet.

MTU, MSS etc...

PPPoE présente un inconvénient considérable, du au fait qu'il encapsule de l'IP avant de le faire transporter sur Ethernet. Cette couche supplémentaire apporte quelques octets de plus dans la trame Ethernet, ce qui est très lourd de conséquences.

Sans précautions particulières, l'internaute risque de rencontrer le fameux problème du "gel de la connexion". Nous allons maintenant analyser dans le détail cet inconvénient et, peut-être, trouver les solutions qui permettraient de supprimer ce désagrément.

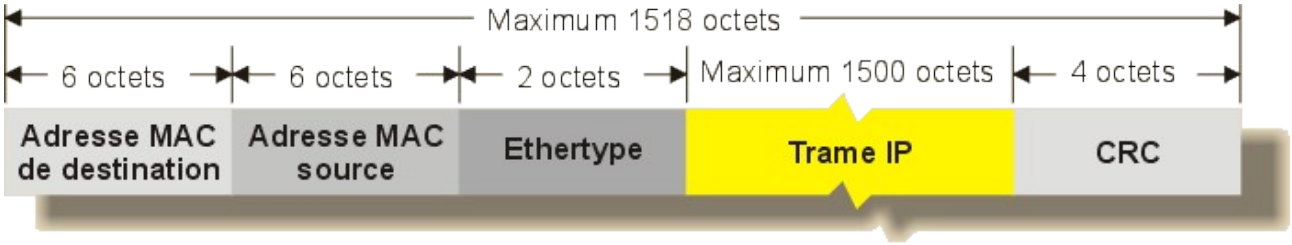
Circulation des données

Une fois encore, nous devrons nous reporter au modèle OSI ou DOD. Rappelez-vous qu'entre les applications qui émettent et reçoivent les données, il y a quelques couches à traverser pour arriver jusqu'au support physique du transport. Lorsqu'une donnée est émise par une application, elle descend les couches une à une, chaque couche ajoutant des informations supplémentaires aux données initiales. Le paquet grossit donc chaque fois qu'il descend d'une couche.

Le problème va venir ici de la couche supplémentaire introduite par PPPoE. Celle-ci ajoute 8 octets supplémentaires, 8 octets de trop dans bien des cas.

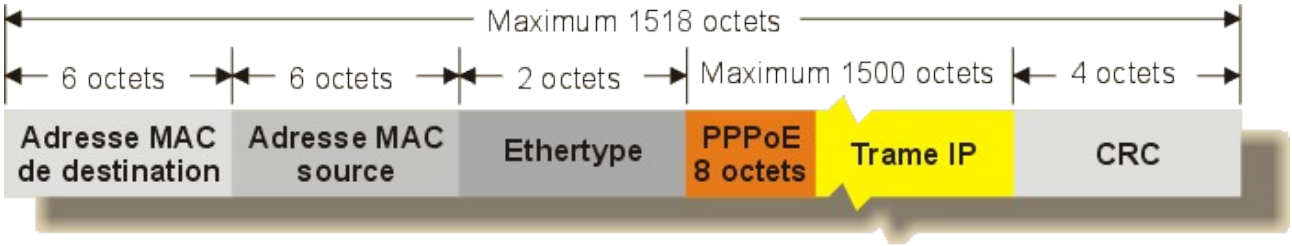
L'IP facile sur Ethernet

Nous le savons, IP est fait pour être transporté par Ethernet. L'Internet a été réalisé à partir de cet axiome.



Comme nous le voyons sur l'illustration, une trame Ethernet fait un maximum de 1518 octets, ce qui laisse à IP 1500 octets maximum. Tout l'Internet est basé sur ce postulat. Je rappelle qu'il s'agit bien ici d'IP sur Ethernet ! Pas de PPPoE dans l'histoire.

PPPoE, l'intrus



Que se passe-t-il lorsque PPPoE s'installe ? Il ajoute, nous l'avons dit, une couche supplémentaire

entre IP qu'il transporte et Ethernet qui le transporte. PPPoE ajoute au total 8 octets supplémentaires. Vous voulez le détail ?

- PPPoE
    - 1 octet pour la version + le type
    - 1 octet de code (voyez RFC[9])
    - 2 octets pour l'identificateur de session
    - 2 octets pour la longueur des données transportées (payload)
- PPP ajoute à ça 2 octets pour indiquer le protocole qu'il transporte

Si l'on part du principe que la trame Ethernet ne peut dépasser 1518 octets (si vous préférez, que la taille maximum du payload Ethernet ne doit pas dépasser 1500 octets), ça veut dire que la trame IP "utile" doit être réduite à 1492 octets.

Compliqué ?

Lorsque les données arrivent sur la couche IP, il se passe le phénomène suivant. IP ne sait pas qu'en dessous de lui il y a PPP et non Ethernet. IP va donc considérer que sa longueur de trame peut atteindre 1500 octets (Maximum Transfert Unit), puisque normalement Ethernet sait transporter un volume de données de cette taille.

Le MTU est fixé, en principe, selon la nature du réseau situé en dessous de IP.

- Lorsque c'est de l'Ethernet, le MTU est 1500.
- Lorsque c'est PPPoE, le MTU doit donc tomber à 1492.
- (Sur du token ring à 16 Mbps, IBM utilise un MTU pourrait aller jusqu'à 17 914 !)

Normalement, lorsque le MTU est le même partout sur les divers composants de l'inter réseau, tout se passe bien.

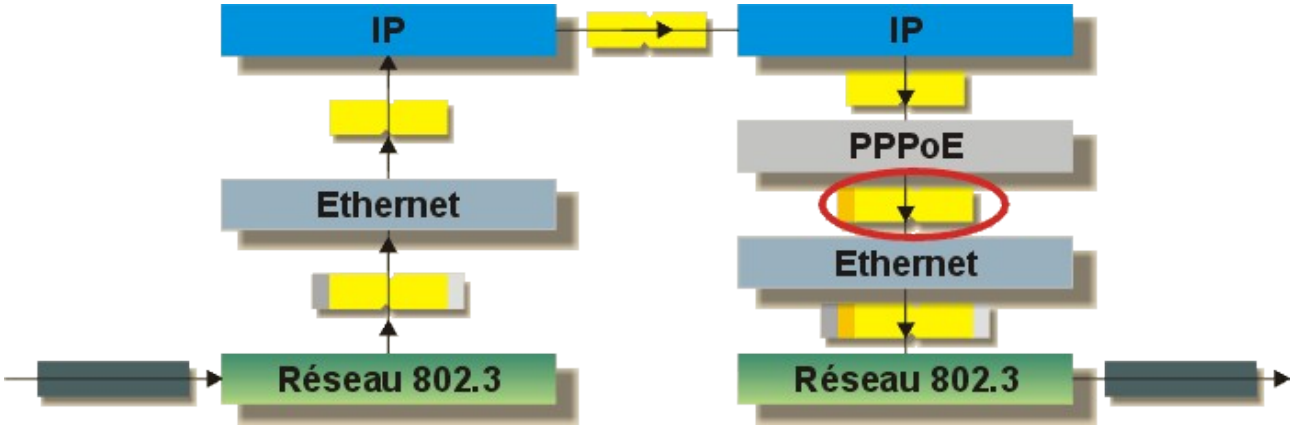
Mais où est le problème ?

Dans une configuration comme celle qui suit. Forcément, quelque part, il va y avoir des configurations comme celle-ci, un routeur avec de l'IP sur Ethernet d'un côté et de l'IP sur PPP sur Ethernet de l'autre. Le paquet jaune représente le datagramme IP. Le point délicat est cerclé de rouge.

Ce paquet, cerclé de rouge, ne doit pas dépasser 1500 octets, à cause des limites d'Ethernet. De ce côté là du routeur, on le sait, mais pas de l'autre.

Si, sur l'entrée (à gauche) le datagramme IP fait 1500 octets, il n'a pas de problèmes pour être véhiculé par Ethernet. Il remonte les couches de gauche et au niveau IP, redescend les couches (à droite). là, il passe par PPPoE qui lui ajoute ses 8 octets. Le paquet devient trop gros pour Ethernet.

9 RFC 2516 en français : http://abcdrfc.free.fr/rfc-vf/rfc2516.html



Dans ce cas, plusieurs éventualités apparaissent.

- Le paquet est fragmenté. En gros, il rentre une seule trame, il en ressort deux.
  Cette méthode présente quelques inconvénients :
  - Le réseau va véhiculer deux fois plus de trames.
  - Il faudra réassembler les fragments à la réception.
- Le paquet est éliminé. Il peut alors se passer deux choses.
  - Un signal ICMP est renvoyé à l'émetteur pour lui signaler que la trame a été rejetée à cause de sa longueur. (Fragmentation needed). L'émetteur va alors recommencer, avec des paquets plus petits, jusqu'à ce que ça passe.
    C'est la méthode "Path MTU Discovery", méthode de découverte du MTU sur un chemin donné.
  - Aucun signal ICMP n'est renvoyé à l'émetteur, parce qu'ICMP est une source de risques pour les routeurs. Beaucoup de firewals bloquent tout trafic ICMP sans discernement (ce qui est une pratique détestable). Dans ce cas, les paquets n'atteindront jamais leur destination, et la source, n'étant pas informée de la raison, finira par clôturer la session.

Finalement, dans un cas concret :

- Vous envoyez une requête à un serveur HTTP (ou FTP). Votre requête arrive jusqu'au serveur.
- Le serveur vous renvoie les données. Ces données sont volumineuses et les datagrammes vont probablement atteindre 1500 octets.
  Lorsque ces données vont être transportées par PPPoE, le volume de données va passer à plus de 1500 octets et le premier équipement qui devra le faire transiter choisira l'une des trois méthodes indiquées ci dessus.
  Si c'est la troisième qui est choisie (on jette sans prévenir personne, ou, du moins, sans pouvoir prévenir personne), la connexion TCP va se geler irrémédiablement.

Et voilà le (sale) travail...

Ce problème peut également apparaître chez vous, si vous partagez votre connexion sur un réseau privé. Votre routeur, quel qu'il soit, sera confronté au même inconvénient. Vous risquez de devoir modifier le MTU de tous les hôtes de votre réseau privé. Nous verrons que si la passerelle est sous Linux et utilise rp-pppoe, le problème pourra être contourné.

Comment résoudre le problème en amont

Grâce à MSS (Maximum Segment Size).

Nous sommes ici au niveau TCP. TCP prépare les paquets de données à envoyer à IP. En agissant sur la taille de ces paquets, on peut éviter les problèmes au niveau du MTU.

Comme l'on sait que les couches inférieures vont ajouter jusqu'à 40 octets aux données, si l'on prend un MSS de 1460 octets, on est à peu près certain que la taille des données transportées au niveau Ethernet ne dépassera pas 1500. Le MSS est établi lors de l'initiation d'une connexion TCP. Le client annonce un MSS typiquement de 1460 dans son premier paquet SYN et le serveur répond dans le SYN/ACK. La trace qui suit est prise sur un poste Windows XP situé sur un LAN connecté à l'internet par une passerelle Sous Debian. Un client FTP (Filezilla), se connecte sur un serveur FTP de l'internet.

```
Frame 1 (62 bytes on wire, 62 bytes captured)
    Arrival Time: Jul 15, 2004 14:04:53.598297000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 62 bytes
    Capture Length: 62 bytes
Ethernet II, Src: 00:05:5d:47:f5:c5, Dst: 00:90:27:71:43:c7
    Destination: 00:90:27:71:43:c7 (Intel_71:43:c7)
    Source: 00:05:5d:47:f5:c5 (D-Link_47:f5:c5)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.0.10 (192.168.0.10), Dst Addr: 195.83.118.1 (195.83.118.1)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 48
    Identification: 0x72da (29402)
    Flags: 0x04 (Don't Fragment)
        0... = Reserved bit: Not set
        .1.. = Don't fragment: Set
        ### Le "flag Don't fragment" est mis, ce qui permet éventuellement de découvrir
        ### le MTU, si tout se passe bien au niveau d'ICMP
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x8de6 (correct)
    Source: 192.168.0.10 (192.168.0.10)
    Destination: 195.83.118.1 (195.83.118.1)
Transmission Control Protocol, Src Port: kpop (1109), Dst Port: ftp (21), Seq: 1814706075, Ack: 0,
Len: 0
    Source port: kpop (1109)
    Destination port: ftp (21)
    Sequence number: 1814706075
    Header length: 28 bytes
    Flags: 0x0002 (SYN)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...0 .... = Acknowledgment: Not set
        .... 0... = Push: Not set
```

```
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 16384
    Checksum: 0xa0e8 (correct)
    Options: (8 bytes)
        Maximum segment size: 1460 bytes
        NOP
        NOP
        SACK permitted
```

Et le serveur répond :

```
Frame 2 (62 bytes on wire, 62 bytes captured)
    Arrival Time: Jul 15, 2004 14:04:53.638354000
    Time delta from previous packet: 0.040057000 seconds
    Time since reference or first frame: 0.040057000 seconds
    Frame Number: 2
    Packet Length: 62 bytes
    Capture Length: 62 bytes
Ethernet II, Src: 00:90:27:71:43:c7, Dst: 00:05:5d:47:f5:c5
    Destination: 00:05:5d:47:f5:c5 (D-Link_47:f5:c5)
    Source: 00:90:27:71:43:c7 (Intel_71:43:c7)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 195.83.118.1 (195.83.118.1), Dst Addr: 192.168.0.10 (192.168.0.10)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 48
    Identification: 0x0000 (0)
    Flags: 0x04 (Don't Fragment)
        0... = Reserved bit: Not set
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 49
    Protocol: TCP (0x06)
    Header checksum: 0x4fc1 (correct)
    Source: 195.83.118.1 (195.83.118.1)
    Destination: 192.168.0.10 (192.168.0.10)
Transmission Control Protocol, Src Port: ftp (21), Dst Port: kpop (1109), Seq: 5846264, Ack:
1814706076, Len: 0
    Source port: ftp (21)
    Destination port: kpop (1109)
    Sequence number: 5846264
    Acknowledgement number: 1814706076
    Header length: 28 bytes
    Flags: 0x0012 (SYN, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x94be (correct)
    Options: (8 bytes)
        Maximum segment size: 1452 bytes
        NOP
        NOP
        SACK permitted
```

Donc nous serons d'accord sur un MSS de 1452 octets, ce qui donnera au sortir de la couche IP un paquet d'un maximum de 1492 octets. On est bon pour PPPoE.

Il est intelligent ce serveur, il a compris tout seul que je me connectais via PPPoE ?

Non, ce n'est pas de l'intelligence, mais un gros bricolage opéré en douce par le client PPPoE de la passerelle. Lorsque le paquet passe par ppp0, le client PPPoE substitue cette valeur de MSS à celle que le serveur a annoncé. Ainsi, le client du LAN agira en conséquence.

Le client PPPoE de Debian est une émanation de RP-PPPoE[10]. La même manipulation est donc faite sur d'autres distributions, qui utilisent RP-PPPoE.

C'est pas très propre, ça ne plaira pas du tout à IPsec, qui trouvera un paquet falsifié et ne le laissera pas passer, mais dans le cas classique d'IP "normal" ça résout le problème sans qu'il soit nécessaire d'intervenir sur tous les postes du LAN, du moins pour TCP. Cette méthode permet de contourner le problème du "path to MTU" qui, comme nous l'avons vu plus haut, ne fonctionne correctement que si ICMP n'est pas bloqué quelque part sur le chemin.

Pour les utilisateurs de GNU/Linux, il existe dans les dernières versions d'IPtables, une cible particulière qui permet de réaliser ce genre de travail[11] sur le MSS. Je ne résiste pas au plaisir de vous citer un extrait de ce HOWTO :

"Ce patch par Marc Boucher, ajoute une nouvelle "target" qui vous permet d'examiner et de changer le MSS dans les paquets TCP SYN, pour contrôler la taille maximum pour cette connexion. Comme l'explique Marc lui même, c'est un hack, utilisé pour résoudre les problèmes engendrés par ces ISPs têtus comme des mules qui bloquent les paquets ICMP Fragmentation Needed à tort."

10 RP-PPPOE : http://christian.caleca.free.fr/pppoe/rp-pppoe_&_linux.htm
11 Cible d'IPtables : http://www.netfilter.org/documentation/HOWTO/fr/netfilter-extensions-HOWTO-4.html#ss4.7