Sniffing et analyse de paquets

1 Outils

Analyser : http://analyzer.polito.it/install/default.htm
RASPPPoE : http://user.cs.tu-berlin.de/~normanb/

2 Prérequis

DSLAM (Digital Subscriber Line Multiplexer) :
Situé sur le réseau de l'opérateur local, au niveau du répartiteur, il fait parti des équipements utilisés pour transformer une ligne téléphonique classique en ligne ADSL permettant la transmission de données, et en particulier l'accès à Internet, à haut débit. La fonction du DSLAM est de regrouper plusieurs lignes ADSL sur un seul support, qui achemine les données en provenance et à destination de ces lignes. (Web Com Agency. Tous droits réservés).

BAS (Broadband Access Server) :
C'est un équipement dont la fonction est de gérer le transport de données en mode ATM dans le cadre des offres d`accès à Internet par ADSL. Sur le réseau de France Télécom, chaque BAS regroupe le trafic ATM issu d`une dizaine de DSLAM. Un BAS gère donc le trafic de l`ensemble des lignes ADSL situées dans les zones couvertes par les DSLAM qui lui sont connectés. La zone ainsi couverte par un BAS est appelée "plaque" par France Télécom. Il est établi un circuit ATM "montant" et un circuit ATM "descendant" entre chaque client connecté et le BAS auquel il est raccordé. (Web Com Agency. Tous droits réservés).

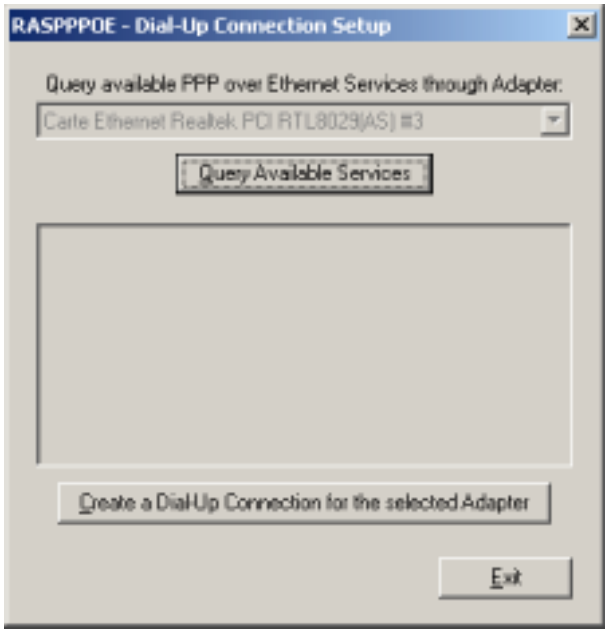
Nous souhaitons analyser les paquets échangés lors d'une requête effectuée avec le pilote *RASPPPoE*, lors de la recherche du BAS auquel le client ADSL doit se connecter.

Nota : le protocole PPPoE n'est utilisé qu'avec des modems ADSL Ethernet.

2.1 Configuration de *RASPPPoE*

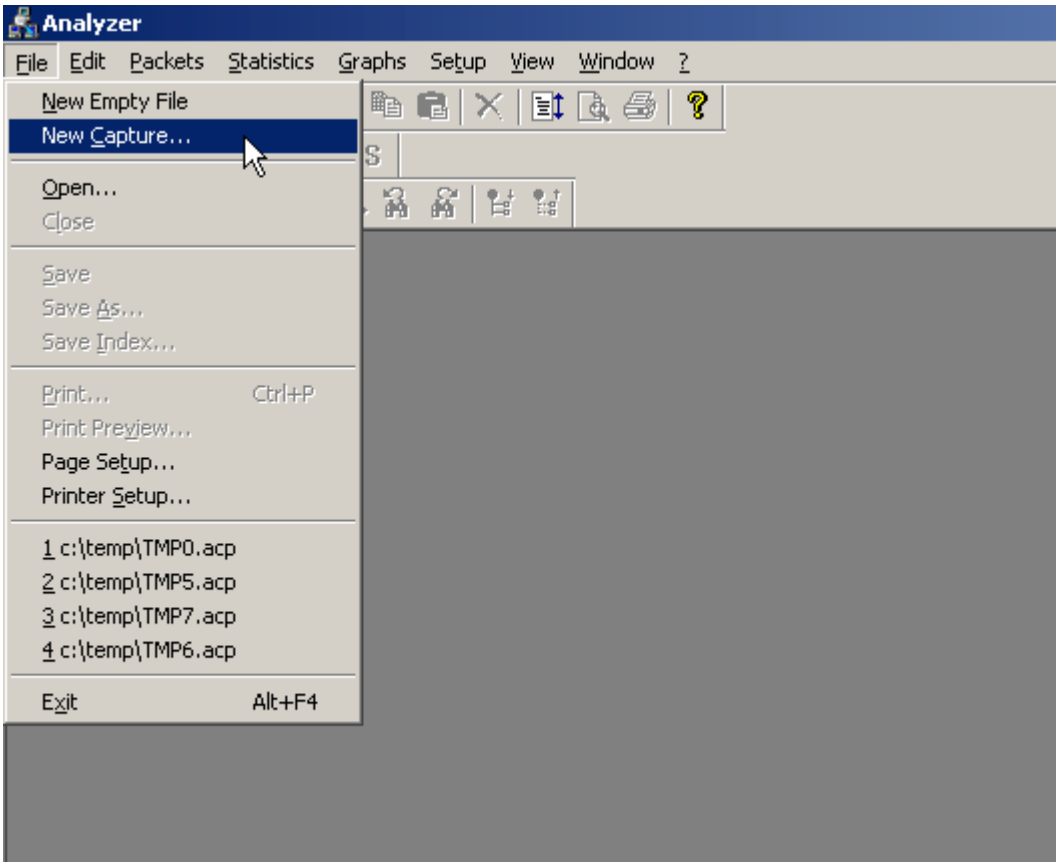
Installez le pilote RASPPPoE comme indiqué dans la documentation.

Lancez *RASPPPoE.exe* sans cliquer sur *Query Available Services*, puisque c'est cette requête que nous allons sniffer.

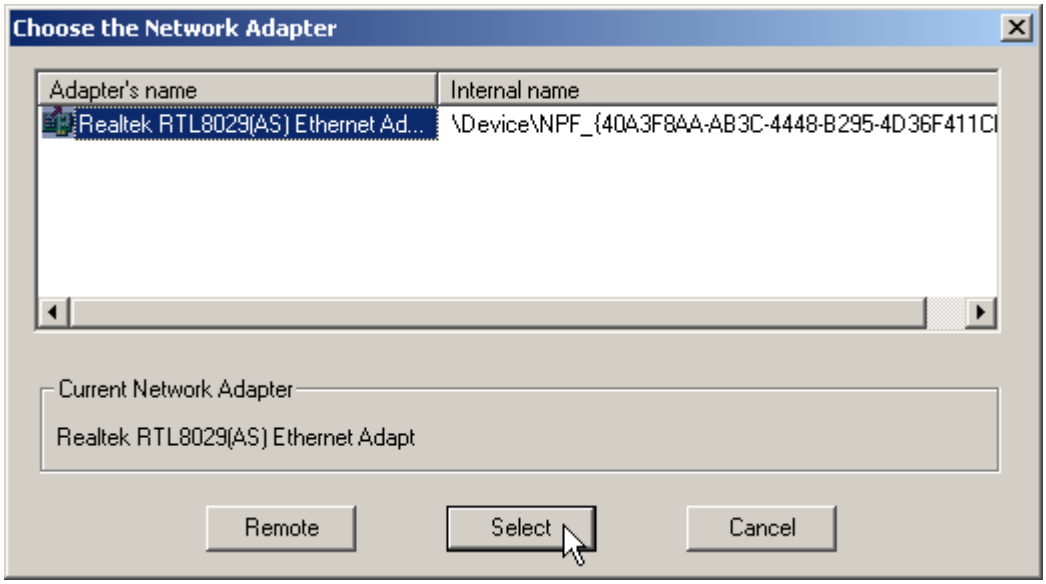


2.2 Configuration et Utilisation de *Analyser*

En parallèle, lancez *Analyser*. Faîtes une nouvelle capture



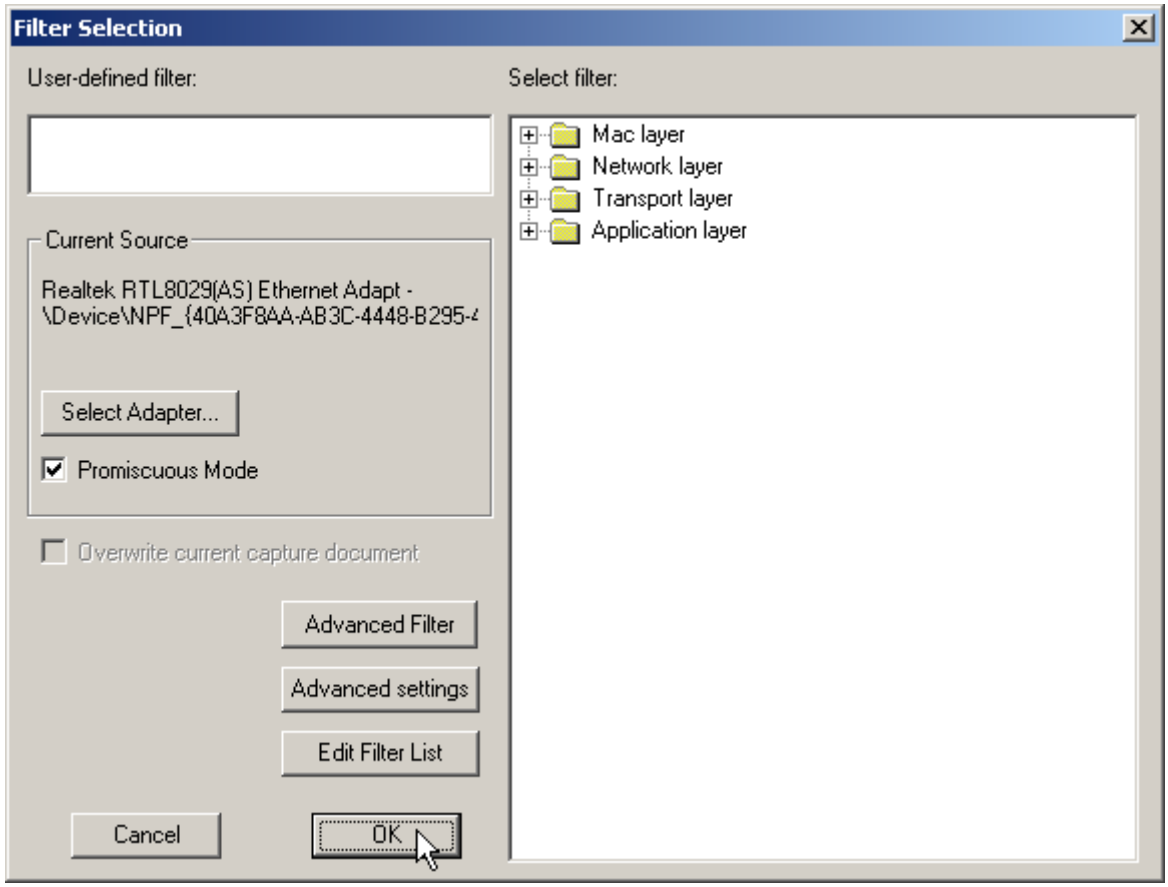
Sélectionnez la carte réseaux à sniffer dans notre cas, celle où est branché le modem).

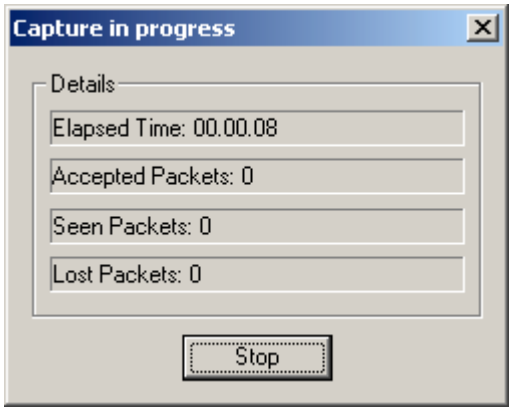


3 Capture

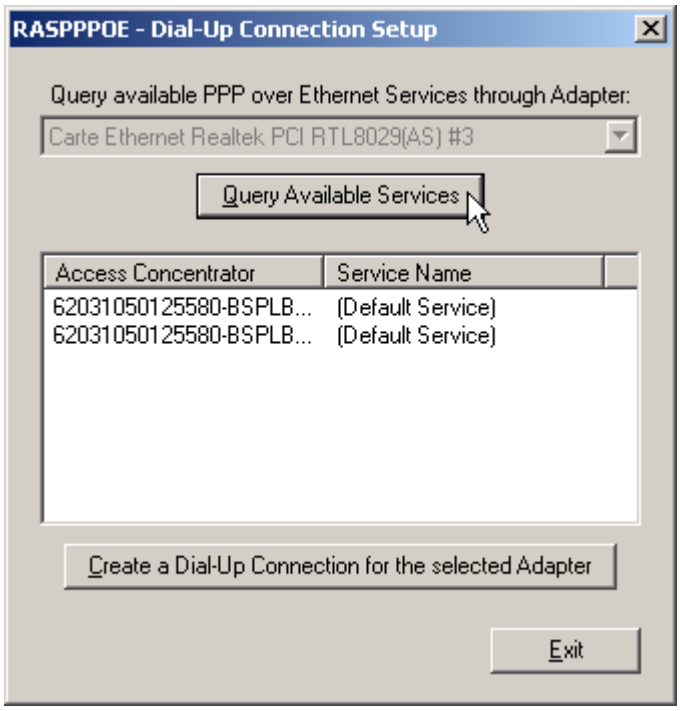
Cliquez sur **OK** dans *Analyser* pour commencer la capture, …

Nota : vous pouvez choisir un filtre pour les paquets à garder. Ici, nous n'en mettrons aucun.

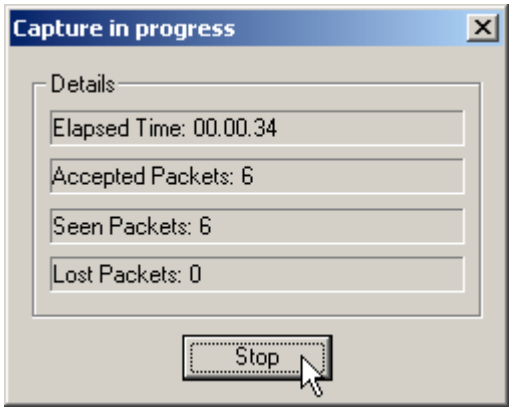




… et cliquez sur *Query Available Services*. Laissez tourner quelques secondes.



Cliquez sur **STOP**.



On peut remarquer que le sniffer a vu 6 paquets passer, et qu'il en a gardé 6 correspondant au filtre appliqué (ici, aucun filtre souhaité).

4 Analyse

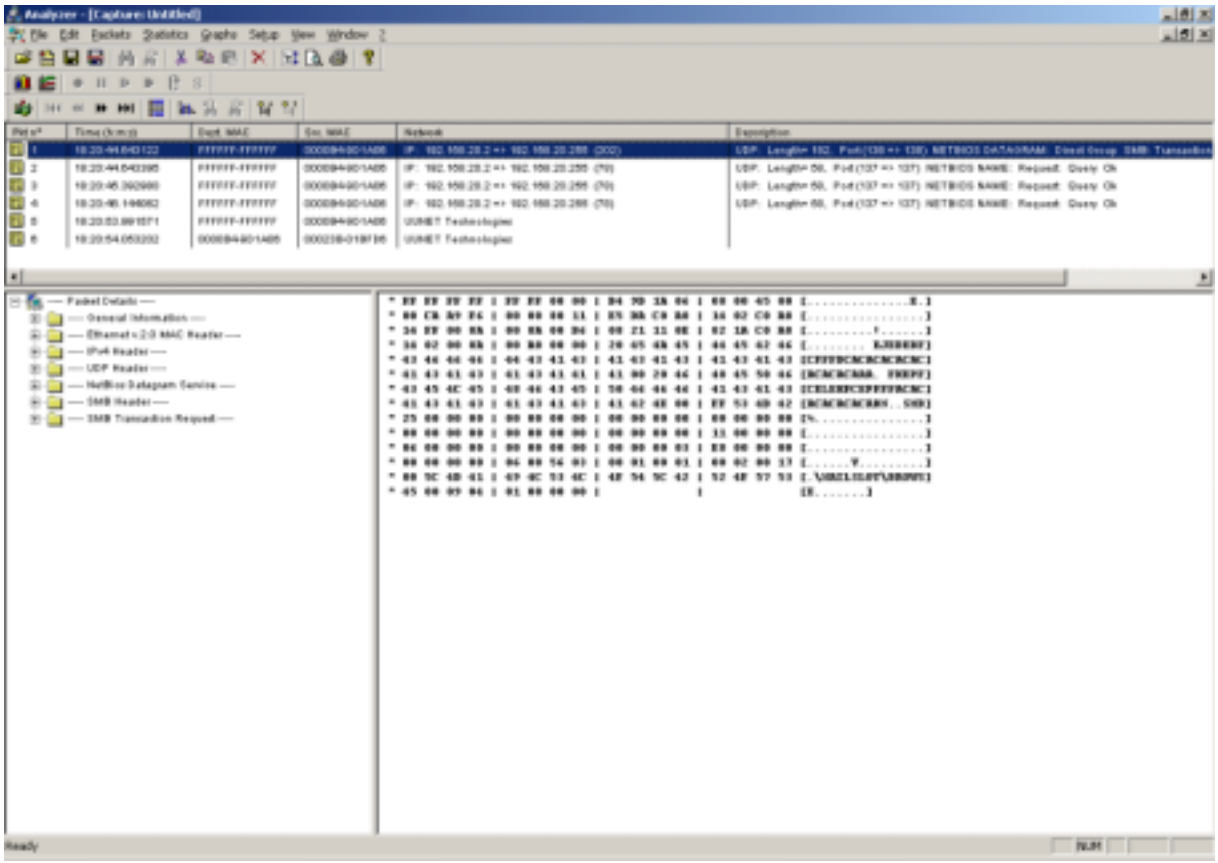
C'est là que nous allons nous servir de notre tête et des RFC.

Nota : Le contenu des paquets apparaît à la fois en héxadécimal et en décimal (entre crochets).

La RFC relative au protocole PPPoE est la RFC 2516 : *A Method for Transmitting PPP Over Ethernet (PPPoE).*

Téléchargeons là sur http://www.ietf.org/rfc/rfc2516.txt, par exemple

4.1 Paquets N°1 à 4



Les 4 premiers paquets sont un broadcast (adresse 192.168.20.255) de la machine 192.168.20.2 sur le réseau local. Le port utilisé (137) indique un accès NetBIOS, donc cette machine souhaite savoir quelles sont les autres machines connectées sur le réseau local.

Ces paquets ne nous intéressent pas.

4.2 Paquet N°5

Jetons un coup d'œil à la RFC .

```
3. Protocol Overview

PPPoE has two distinct stages.  There is a Discovery stage and a PPP Session stage.
```

Et plus loin :

```
   An Ethernet frame is as follows:
                                 1
                 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |       DESTINATION_ADDR        |
                |          (6 octets)           |
                |                               |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |         SOURCE_ADDR           |
                |          (6 octets)           |
                |                               |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |    ETHER_TYPE  (2 octets)     |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                ~                               ~
                ~           payload             ~
                ~                               ~
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |           CHECKSUM            |
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   The DESTINATION_ADDR field contains either a unicast Ethernet
   destination address, or the Ethernet broadcast address (0xffffffff).
   For Discovery packets, the value is either a unicast or broadcast
   address as defined in the Discovery section.  For PPP session
   traffic, this field MUST contain the peer's unicast address as
   determined from the Discovery stage.

   The SOURCE_ADDR field MUST contains the Ethernet MAC address of the
   source device.

   The ETHER_TYPE is set to either 0x8863 (Discovery Stage) or 0x8864
   (PPP Session Stage).

   The Ethernet payload for PPPoE is as follows:

                       1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  VER  | TYPE  |      CODE     |          SESSION_ID           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            LENGTH             |           payload             ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   The VER field is four bits and MUST be set to 0x1 for this version of
   the PPPoE specification.

   The TYPE field is four bits and MUST be set to 0x1 for this version
   of the PPPoE specification.

   The CODE field is eight bits and is defined below for the Discovery
   and PPP Session stages.

   The SESSION_ID field is sixteen bits.  It is an unsigned value in
   network byte order.  It's value is defined below for Discovery
   packets.  The value is fixed for a given PPP session and, in fact,
   defines a PPP session along with the Ethernet SOURCE_ADDR and
   DESTINATION_ADDR.  A value of 0xffff is reserved for future use and
   MUST NOT be used

   The LENGTH field is sixteen bits.  The value, in network byte order,
   indicates the length of the PPPoE payload.  It does not include the
   length of the Ethernet or PPPoE headers.
```

Analysons le paquet n°5 :

```
* FF FF FF FF | FF FF 00 00 | B4 9D 1A 06 | 88 63 11 09 [............c..]
* 00 00 00 18 | 01 01 00 00 | 01 03 00 10 | 52 53 50 45 [............RSPE]
* 00 00 00 00 | F0 FF 25 92 | F2 E7 C2 01 |             [......%.....]
```

DESTINATION_ADDR : adresse de broadcast
SOURCE_ADDR : adresse MAC de la carte réseaux configurée en 192.168.20.2
ETHER_TYPE : discovery stage
VER (4 bits soient un demi octet)
TYPE (4 bits)
CODE (8 bits = 1 octet)
SESSION_ID (16 bits = 2 octets
LENGTH (16 bits = 2 octets) ici, la longueur de la charge est de 0x0018 bits soient 24 bits (3 octets) en décimal.

Or :

```
5.1 The PPPoE Active Discovery Initiation (PADI) packet

   The Host sends the PADI packet with the DESTINATION_ADDR set to the
   broadcast address.  The CODE field is set to 0x09 and the SESSION_ID
   MUST be set to 0x0000.
```

Nous somme ici dans le cas d'un paquet appelé PADI, puisque nous avons bien l'adresse de broadcast, un identifiant de session à 0x0000 et un CODE à 0x09

Analysons alors la charge (payload) et le reste du paquet :

La RFC nous dit :

```
   The PPPoE payload contains zero or more TAGs.  A TAG is a TLV (type-
   length-value) construct and is defined as follows:

                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          TAG_TYPE             |        TAG_LENGTH             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          TAG_VALUE ...                                        ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   TAG_TYPE is a sixteen bit field in network byte order.  Appendix A
   contains a list of all TAG_TYPEs and their TAG_VALUEs.

   TAG_LENGTH is a sixteen bit field.  It is an unsigned number in
   network byte order, indicating the length in octets of the TAG_VALUE.

   If a discovery packet is received with a TAG of unknown TAG_TYPE, the
   TAG MUST be ignored unless otherwise specified in this document.
   This provides for backwards compatibility if/when new TAGs are added.
   If new mandatory TAGs are added, the version number will be
   incremented.

   Some example Discovery packets are shown in Appendix B.
```

```
*             | 01 01 00 00 | 01 03 00 10 | 52 53 50 45 [    ........RSPE]
* 00 00 00 00 | F0 FF 25 92 | F2 E7 C2 01 |             [......%.....]
```

TAG_TYPE : 0x0101 (Service-Name)
TAG_LENGTH : 0x0000
TAG_VALUE : il n'y en a pas car TAG_LENGTH = 0 octet

TAG_TYPE : 0x0103 (Host-Uniq)
TAG_LENGTH : 0x0010 (= 16 octets une fois converti en décimal)
TAG_VALUE : 52 53 50 45 00 00 00 00 F0 FF 25 92 F2 E7 C2 01

Nota : le TAG_TYPE 0x0000 End-Of-List est facultatif, mais conseillé pour une compatibilité future (c.f. RFC 2516 dans Appendix A, *TAG_TYPES and TAG_VALUES*, page 10).

4.3 Paquet N°6

```
* 00 00 B4 9D | 1A 06 00 02 | 3B 01 BF D6 | 88 63 11 07 [........;....c..]
* 00 00 00 37 |
```

DESTINATION_ADDR
SOURCE_ADDR
ETHER_TYPE : 0x8863 (C'est toujours un paquet dans la *discovery stage*)
VER : 0x1
TYPE : 0x1
CODE : 0x07
SESSION_ID : 0x0000
LENGTH : 0x0037

```
*             01 01 00 00 | 01 03 00 10 | 52 53 50 45 [...7........RSPE]
* 00 00 00 00 | F0 FF 25 92 | F2 E7 C2 01 | 01 02 00 17 [......%.........]
* 36 32 30 33 | 31 30 35 30 | 31 32 35 35 | 38 30 2D 42 [62031050125580-B]
* 53 50 4C 42 | 31 30 38 01 | 01 00 00    |             [SPLB108....]
```

TAG_TYPE : 0x0101 (Service-Name)
TAG_LENGTH : 0x0000
TAG_VALUE : il n'y en a pas car TAG_LENGTH = 0 octet

TAG_TYPE : 0x0103 (Host-Uniq)
TAG_LENGTH : 0x0010 (= 16 octets une fois converti en décimal)
TAG_VALUE : 52 53 50 45 00 00 00 00 F0 FF 25 92 F2 E7 C2 01

TAG_TYPE : 0x0102 (AC-Name)
TAG_LENGTH : 0x0017 (= 23 octets une fois converti en décimal)
TAG_VALUE : 36 32 30 33 31 30 35 30 31 32 35 35 38 30 2D 42 53 50 4C 42 31 30 38 01 01 00 00

Si on se réfère à la RFC :

> The PADO packet MUST contain one AC-Name TAG containing the Access Concentrator's name, a Service-Name TAG identical to the one in the PADI, and any number of other Service-Name TAGs indicating other services that the Access Concentrator offers. If the Access Concentrator can not serve the PADI it MUST NOT respond with a PADO.

D'où le nom de l'*Access Concentrator's name* :

62031050125580-BSPLB108

Selon l'appellation de France Télécom, il s'agit du BAS du PLessis-Bouchard N°108

5 Conclusion

Cette analyse de paquets à l'aide des RFC se déroule toujours de la même façon.