

**- TP 3 -**

# **Protocole de Couche Liaison PPP**

*par [REDACTED] & Édouard Lumet*

## Sommaire

1. Mise en place de la maquette.....	3
2. Mise en place d'un serveur PPP.....	3
3. Mise en place et connexion d'un client PPP.....	5
4. Étude de fonctionnement du protocole PPP.....	6

## 1. Mise en place de la maquette

Dans un premier temps, nous avons relié nos machines au réseau de l'IUT pour télécharger quelques paquets :

```
iutrt2A@PC- ~ $ sudo apt-get install pppoe pppoeconf wireshark
```

Ensuite, nous avons câblé les deux machines sur un même switch afin qu'elles ne puissent communiquer uniquement entre elles.

## 2. Mise en place d'un serveur PPP

- Le fichier `pppoe-server-options` n'existe pas dans `/etc/ppp`, nous avons dû le rajouter avec la commande `touch /etc/ppp/pppoe-server-options`. Ce fichier permet de configurer les options de notre serveur PPP.
- Le fichier `/etc/ppp/pap-secrets` permet d'utiliser une authentification avec PAP (login et mot de passe). Dans la partie INBOUND CONNECTIONS de ce fichier, nous ajoutons la ligne suivante : `iutrt2A* « rt2A » *`

Ceci permet d'ajouter l'utilisateur « iutrt2A » ayant pour mot de passe « rt2A ».

- Pour lancer le serveur PPP sur notre machine nous utilisons le script suivant :

```
#!/bin/bash
#nom du serveur ppp
PROV=PPPoE7
#nombre de client maximum
MAX=10
#début de la plage attribuable par le serveur
BASE=10.2.221.2
#adresse ip publique du serveur ppp
MYIP=10.2.221.1
#mise en place de l'ip forwarding
echo "1" > "/proc/sys/net/ipv4/ip_forward"
#lancement du serveur ppp sur l'interface eth0
/usr/sbin/pppoe-server -L $MYIP -T 60 -I eth0 -N $MAX -C $PROV -S $PROV -R $BASE
```

- -R permet de fixer l'adresse IP distante de départ. « pppoe-server » conserve automatiquement la trace du groupe d'adresses et transmet une adresse IP distante valide à pppd.
- La commande utilisée pour mettre les droits d'exécution au script précédent est : `sudo chmod +x /etc/init.d/serveurppp`

- Le paramètre MRU ( Unité de Réception Maximum ) permet au pppd de demander de se faire renvoyer les paquets pour au plus N octets du MRU.

Le paramètre MTU ( Unité de Transmission Maximum ) permet au pppd de demander que les codes réseaux du noyau envoie des paquets d'au plus N octets dans l'interface réseau PPP.

- On lance enfin le serveur avec la commande **bash /etc/init.d/serveurppp**

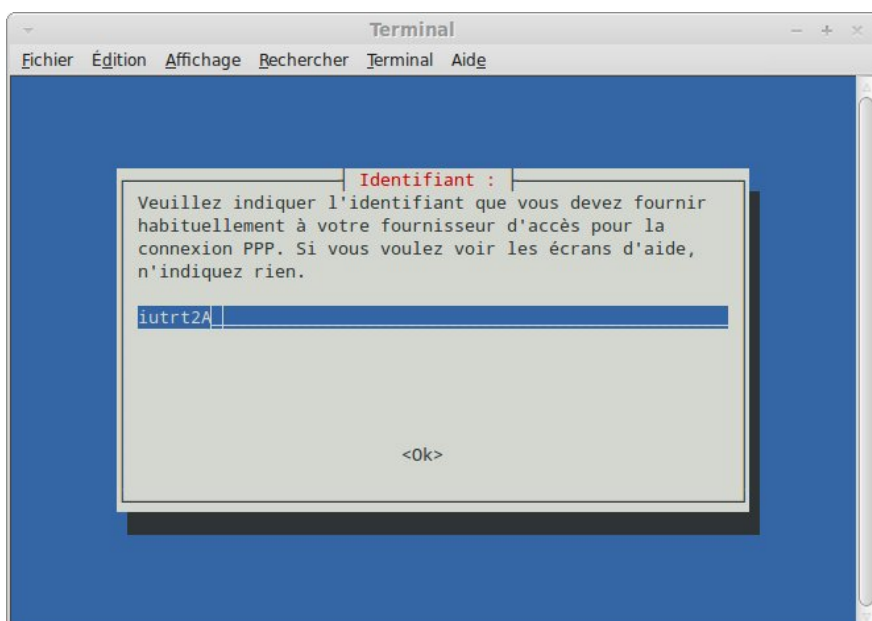
### 3. Mise en place et connexion d'un client PPP

- Sur les deux machines, on repère notre carte réseau physique *eth0* ainsi qu'une carte virtuelle nommée *ppp0*, comme ceci :

```
eth0  Link encap:Ethernet HWaddr 2c:27:d7:24:0d:ce
      inet adr:192.168.0.26 Bcast:192.168.0.255 Masque:255.255.255.0
      adr inet6: fe80::2e27:d7ff:fe24:dce/64 Scope:Lien
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      Packets reçus:46636 erreurs:0 :0 overruns:0 frame:0
      TX packets:15897 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 lg file transmission:1000
      Octets reçus:28209771 (28.2 MB) Octets transmis:1809124 (1.8 MB)
      Interruption:19 Mémoire:f3200000-f3220000

ppp0  Link encap:Protocole Point-à-Point
      inet adr:10.2.221.2 P-t-P:10.2.221.1 Masque:255.255.255.255
      UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1492 Metric:1
      Packets reçus:4 erreurs:0 :0 overruns:0 frame:0
      TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 lg file transmission:3
      Octets reçus:46 (46.0 B) Octets transmis:52 (52.0 B)
```

- Les différentes étapes de l'initialisation de connexion PPP sont les suivantes : le client envoie un **configure-request** avec diverses options à négocier, l'authentification, la compression, etc. Ensuite, les options peuvent être partiellement ou entièrement rejetées, ou bien être toutes acceptées (resp. **configuration-nack**, **configuration-reject**, **configuration-ack**).
- La commande **pppoeconf** modifie le fichier de configuration */etc/ppp/peers/dsl-provider*. C'est la commande à utiliser pour initialiser la connexion PPPoE.



- Comme nous l'avons vu sur l'illustration précédente, deux cartes réseaux sont visibles. L'interface PPP possède l'adresse 10.2.221.2, attribuée par le serveur PPP. Elle figurait dans les adresses du fichier de configuration de PPP.
- Les commandes **pon**, **poff** et **plog** permettent respectivement de démarrer, arrêter et lister les connexions PPP.

## 4. Étude de fonctionnement du protocole PPP

```

 8 *REF*      HewlettP_24:0d:ce HewlettP_20:68:87 PPP LCP      36 Configuration Request
 9 0.000672   HewlettP_20:68:87 HewlettP_24:0d:ce PPP LCP      60 Configuration Ack
10 0.000794
11 0.000879
12 0.000882
13 0.001070   HewlettP_24:0d:ce HewlettP_20:68:87 PPP IPCP     44 Configuration Request
14 0.001120
15 0.001158
16 0.001378
17 0.001608   HewlettP_20:68:87 HewlettP_24:0d:ce PPP IPCP     60 Configuration Reject
18 0.001809   HewlettP_24:0d:ce HewlettP_20:68:87 PPP IPCP     32 Configuration Request
19 0.002449   HewlettP_20:68:87 HewlettP_24:0d:ce PPP IPCP     60 Configuration Nak
20 0.002646   HewlettP_24:0d:ce HewlettP_20:68:87 PPP IPCP     32 Configuration Request
21 0.005106   HewlettP_20:68:87 HewlettP_24:0d:ce PPP IPCP     60 Configuration Ack
22 5.837480   HewlettP_20:68:87 HewlettP_24:0d:ce PPP LCP      60 Echo Request
23 5.837686   HewlettP_24:0d:ce HewlettP_20:68:87 PPP LCP      30 Echo Reply
24 7.166983   HewlettP_24:0d:ce HewlettP_20:68:87 PPP LCP      30 Echo Request
25 7.167874   HewlettP_20:68:87 HewlettP_24:0d:ce PPP LCP      60 Echo Reply

```

Échanges PPP client (:ce)/serveur (:87)

- Pour initialiser la connexion PPP, le protocole LCP est utilisé par l'envoi d'un **configuration-request**. Ici, les paramètres sont acceptés, un **configuration-ack** est envoyé en réponse par le serveur :

```

 8 *REF*      HewlettP_24:0d:ce HewlettP_20:68:87 PPP LCP      36 Configuration Request
 9 0.000672   HewlettP_20:68:87 HewlettP_24:0d:ce PPP LCP      60 Configuration Ack

```

- L'authentification de l'utilisateur s'est fait ici via PAP (voir partie 2). Les échanges ne sont donc pas sécurisés.
- Enfin, le protocole IPCP est utilisé pour la mise en place de la liaison. Ici, le serveur renvoie un **configure-reject** car les paramètres DNS sont non négociables :

```

17 0.001608   HewlettP_20:68:87 HewlettP_24:0d:ce PPP IPCP     60 Configuration Reject

```

Frame 17: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)  
 Ethernet II, Src: HewlettP\_20:68:87 (2c:27:d7:20:68:87), Dst: HewlettP\_24:0d:ce (2c:27:d7:24:0d:ce)  
 PPP-over-Ethernet Session  
 Point-to-Point Protocol  
 PPP IP Control Protocol  
 Code: Configuration Reject (4)  
 Identifier: 1 (0x01)  
 Length: 16  
 Options: (12 bytes), Primary DNS Server IP Address, Secondary DNS Server IP Address  
 ▶ Primary DNS Server IP Address: 0.0.0.0  
 ▶ Secondary DNS Server IP Address: 0.0.0.0

Le client renvoie donc un **configure-request** sans ces paramètres mais le serveur renvoie un **configuration-nak** car l'option proposée n'est pas satisfaisante. Il lui fait une proposition en même temps :

```

19 0.002449   HewlettP_20:68:87 HewlettP_24:0d:ce PPP IPCP     60 Configuration Nak

```

Frame 19: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)  
 Ethernet II, Src: HewlettP\_20:68:87 (2c:27:d7:20:68:87), Dst: HewlettP\_24:0d:ce (2c:27:d7:24:0d:ce)  
 PPP-over-Ethernet Session  
 Point-to-Point Protocol  
 PPP IP Control Protocol  
 Code: Configuration Nak (3)  
 Identifier: 2 (0x02)  
 Length: 10  
 Options: (6 bytes), IP address  
 ▶ IP address: 10.2.221.8

Le client renvoie un dernier **configuration-request** avec le paramètre proposé par le serveur, ce dernier acceptant enfin le tout par un **configuration-ack** :

```

20 0.002646 HewlettP_24:0d:ce HewlettP_20:68:87 PPP IPCP 32 Configuration Request
Frame 20: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)
Ethernet II, Src: HewlettP_24:0d:ce (2c:27:d7:24:0d:ce), Dst: HewlettP_20:68:87 (2c:27:d7:20:68:87)
PPP-over-Ethernet Session
Point-to-Point Protocol
PPP IP Control Protocol
Code: Configuration Request (1)
Identifier: 3 (0x03)
Length: 10
Options: (6 bytes), IP address
  IP address: 10.2.221.8
    
```

- La composition d'une trame PPPoE est la suivante :

**Données**

Type	Length	Value
16 bits	16 bits	

**PPPoE**

Version	Type	Code	Session Identifier	Length	Données	FCS
4 bits	4 bits	8 bits	16 bits	16 bits		