

# RFC 1661 : The Point-to-Point Protocol (PPP)

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 avril 2007

Date de publication du RFC : Juillet 1994

<https://www.bortzmeyer.org/1661.html>

---

L'incroyable protocole PPP assure toujours une grande partie des connexions Internet dans le monde. Ce RFC relativement court décrit le fonctionnement de PPP.

PPP, comme son nom l'indique, sert aux liaisons point-à-point, par opposition aux réseaux partagés comme Ethernet. Initialement conçu pour remplacer SLIP sur les liaisons modem ou les liaisons série à très courte distance, ainsi que pour remplacer différents protocoles privés entre routeurs sur les WAN, il continue aujourd'hui sa carrière même sur ADSL avec PPPoE (décrit dans le RFC 2516<sup>1</sup>).

PPP est dérivé de HDLC mais est considérablement plus simple. Il comprend trois parties :

- Une méthode d'encapsulation des trames, dans la section 2 de notre RFC, proche de celle de HDLC,
- Un sous-protocole, LCP ("*Link Control Protocol*"), de contrôle de la liaison, décrit dans le reste du RFC,
- Et plusieurs protocoles de contrôle des protocoles réseaux situés plus haut, les NCP ("*Network Control Protocol*"). PPP travaille à la couche 2 et il est donc indépendant du protocole utilisé en couche 3, à part chaque NCP, qui, lui, dépend du protocole de couche 3. Il existe donc un NCP pour IPv4, IPCP, décrit dans le RFC 1332, un NCP pour IPv6, IPv6CP, décrit dans le RFC 5072, etc.

Dans les exemples de configuration PPP suivants, je prendrai comme référence le programme **pppd**, utilisé sur beaucoup de systèmes Unix, notamment Linux (les utilisateurs de FreeBSD préfèrent en général "*Userland PPP*" <[http://www.freebsd.org/doc/en\\_US.ISO8859-1/books/handbook/userppp.html](http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/userppp.html)>). Les messages affichés sont extraits du journal de pppd.

Voici par exemple la négociation LCP qui permet aux deux pairs de se mettre d'accord sur les options de la ligne (un gros avantage de PPP par rapport à SLIP) :

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc2516.txt>

```
Apr 22 09:30:01 ludwigVI pppd[3055]: rcvd [LCP ConfReq id=0x3e <mru 1492> <auth chap MD5> <magic 0x7de8a7c3>
Apr 22 09:30:01 ludwigVI pppd[3055]: sent [LCP ConfReq id=0x1 <mru 1400> <asyncmap 0x0> <magic 0x6a0c5efc>
Apr 22 09:30:01 ludwigVI pppd[3055]: sent [LCP ConfAck id=0x3e <mru 1492> <auth chap MD5> <magic 0x7de8a7c3>
Apr 22 09:30:01 ludwigVI pppd[3055]: rcvd [LCP ConfRej id=0x1 <asyncmap 0x0> <pcomp> <accomp>]
Apr 22 09:30:01 ludwigVI pppd[3055]: sent [LCP ConfReq id=0x2 <mru 1400> <magic 0x6a0c5efc>]
Apr 22 09:30:01 ludwigVI pppd[3055]: rcvd [LCP ConfAck id=0x2 <mru 1400> <magic 0x6a0c5efc>]
```

On notera que certaines options ont été rejetées par un des pairs (par exemple la compression `pcomp`).

PPP permet d'avoir une seule authentification pour tous les protocoles réseaux, gérée par LCP, et reposant en général sur les protocoles PAP ou CHAP.

On voit ici le dialogue CHAP. La machine a reçu un défi CHAP envoyé par le BAS et répond avec un NAI (RFC 7542) configuré par l'option `name` :

```
Apr 22 09:30:01 ludwigVI pppd[3055]: rcvd [CHAP Challenge id=0x1 <35de4b970986e23fc4447ee4416beada>, name =
Apr 22 09:30:01 ludwigVI pppd[3055]: sent [CHAP Response id=0x1 <c43b7ad1c1cdc8aaa74273efdebb3af0>, name =
```

LCP permet de définir des tests de la ligne (section 5.8 du RFC). On les active avec les options :

```
lcp-echo-interval 120
lcp-echo-failure 3
```

et on peut voir qu'ils sont régulièrement effectués :

```
Apr 22 09:30:02 ludwigVI pppd[3055]: sent [LCP EchoReq id=0x0 magic=0xa1ce4573]
Apr 22 09:30:02 ludwigVI pppd[3055]: rcvd [LCP EchoRep id=0x0 magic=0xe8d1ffb5]
```

PPP permet de configurer la MRU ("*Maximum Receive Unit*") de la ligne, souvent une source d'ennuis importante (voir par exemple le RFC 4638). Avec `pppd`, c'est :

```
mtu 1492
mru 1400
```

et cela donne, dans la négociation :

```
Apr 22 09:30:01 ludwigVI pppd[3055]: sent [LCP ConfReq id=0x1 <mru 1400> <asyncmap 0x0> <magic 0x6a0c5efc> <
```

`pppd` et, bien sûr, le protocole PPP lui-même dispose de nombreuses autres options.