Introducció al *Point-to-Point Protocol* (PPP)

Llorenç Cerdà Alabern.

1 Introducció

Aquesta pràctica il·lustra el funcionament del *Point-to-Point Protocol* (PPP) que es fa servir en Internet. Per això s'estableix una connexió PPP entre dues màquines LINUX a través del port sèrie mitjançant la comanda `pppd`.

2 Descripció del protocol PPP

Objectiu: transport de datagrames multi-protocol en un enllaç punt-a-punt (figura 1).

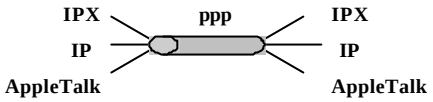


figura 1: Multiplexat de datagrames multiprotocol en un enllaç PPP.

El protocol defineix mecanismes per: (RFC-1661)

- Encapsulat (i multiplexat) de datagrames multiprotocol.
- Establiment, configuració i test de la connexió (*Link Control Protocol*, LCP).
- Una família de protocols per establir i configurar diferents protocols de nivell de xarxa (*Network Control Protocol*, NCP). En el cas de IP es diu *Internet Protocol Control Protocol*, IPCP.
- Opcionalment, PPP suporta un protocol de compressió de paquets (*Compression Control Protocol*, CCP), RFC-1962. Les trames intercanviades en la negociació del protocol CCP son anàlogues a les del protocol LCP. En aquesta la negociació s'acorda l'algorisme de compressió que es farà sevir en l'enllaç, com ara "deflate 15", "bsd-v1 15" o altres.

La següent figura mostra el diagrama d'estats corresponent a l'establiment i terminació d'un enllaç PPP.

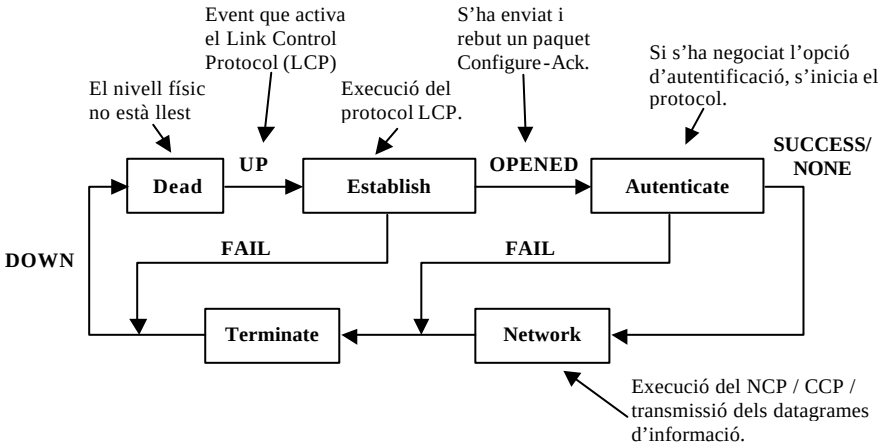


figura 2: Diagrama d'estats d'un enllaç PPP.

2.1 Format d'una trama PPP

La figura 3 mostra el format d'una trama PPP. Aquest consta d'un camp de protocol de dos bytes i un de dades. El camp de protocol identifica el protocol al qual correspon la informació encapsulada en el camp de dades.

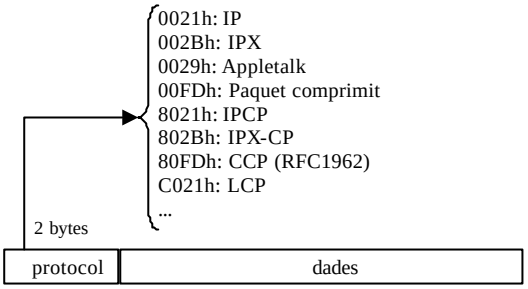


figura 3: Trama PPP.

2.2 Transmissió de trames PPP en diferents tipus d'enllaços

La figura 4 mostra enllaços típics on es fa servir PPP. Existeixen diferents RFCs que expliquen com transmetre les trames PPP segons la xarxa física que es fa servir:

- Tipus HDLC (RFC-1549).
- X.25 (RFC-1598)
- ISDN (RFC-1618)
- Frame Relay (RFC-1973)
- ATM (RFC-2364)
- ...

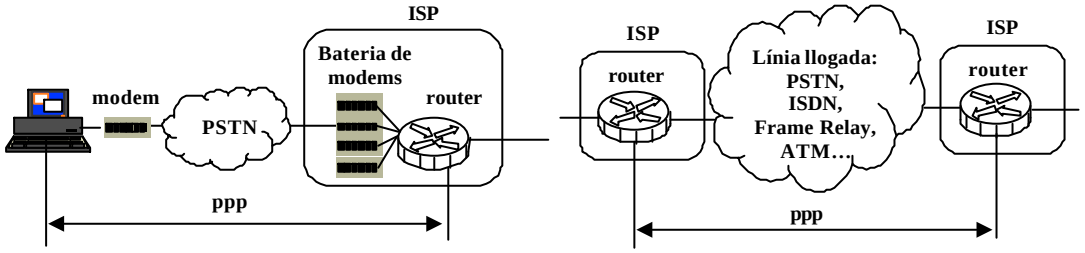


figura 4: Usos típics de PPP.

2.2.1 PPP en enllaços tipus HDLC (RFC-1549)

S'utilitza p.e. en enllaços a través del port sèrie RS232 fent servir modems o un cable modem null. Aplicable en enllaços síncrons/asíncrons orientats a bit/caràcter. La següent figura mostra l'encapsulament.

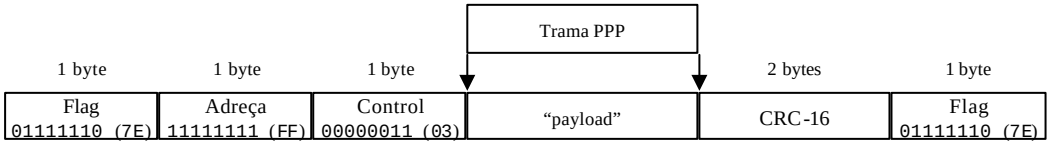


figura 5: Encapsulament de trames PPP en enllaços HDLC.

Pel mecanisme de transparència es fa servir una de les següents tècniques:

***Bit stuffing*:** s'insereix un '0' cada 5 '1'.

Character stuffing: el codi d'escape és 7Dh. Als codis escapats es complementa el bit 6 (fent un XOR amb 20h). Això es fa perquè els codis inferiors a 20h són codis de control d'alguns modems. La màscara de 32 bits ACCM identifica quins codis han d'escapar-se (cada bit de la màscara identifica un dels primers 32 codis). Per exemple, ACCM = FFFFFFFFh vol dir escapar els 32 primers codis, ACCM = 0 vol dir no escapar-ne cap.

2.3 *Link Control Protocol,* LCP (RFC-1661)

La figura 6 mostra el format de les trames que fa servir LCP per a la negociació de les opcions de l'enllaç PPP. El camp "id" de la trama LCP permet establir la correspondència entre una petició i la resposta corresponent.

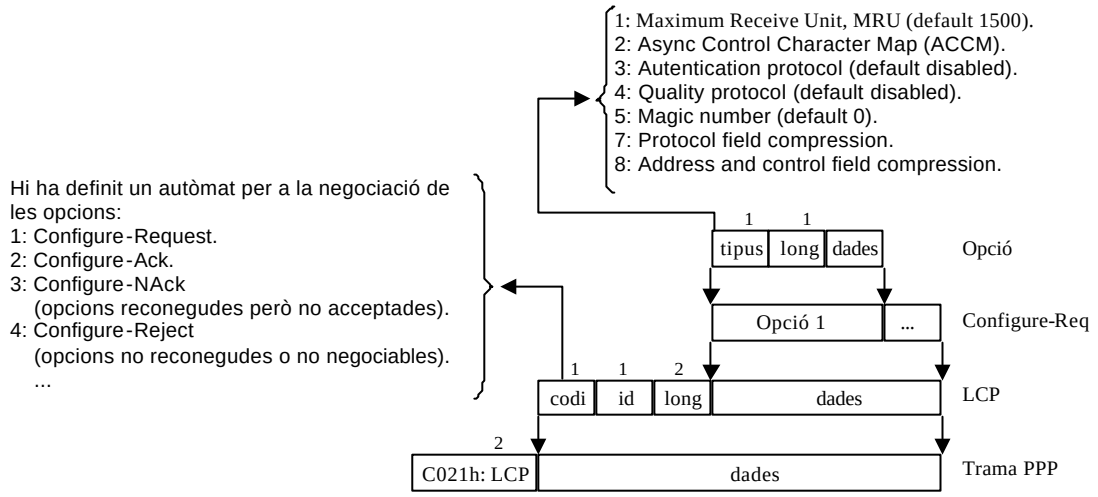


figura 6: Format i encapsulat d'una trama PPP amb informació LCP.

2.4 *PPP Internet Protocol Control Protocol*, IPCP (RFC-1172)

És el NCP que s'utilitza per a configurar la transmissió de paquets IP sobre un enllaç PPP. Fa servir el mateix autòmat per a la negociació d'opcions i format de les trames que LCP:

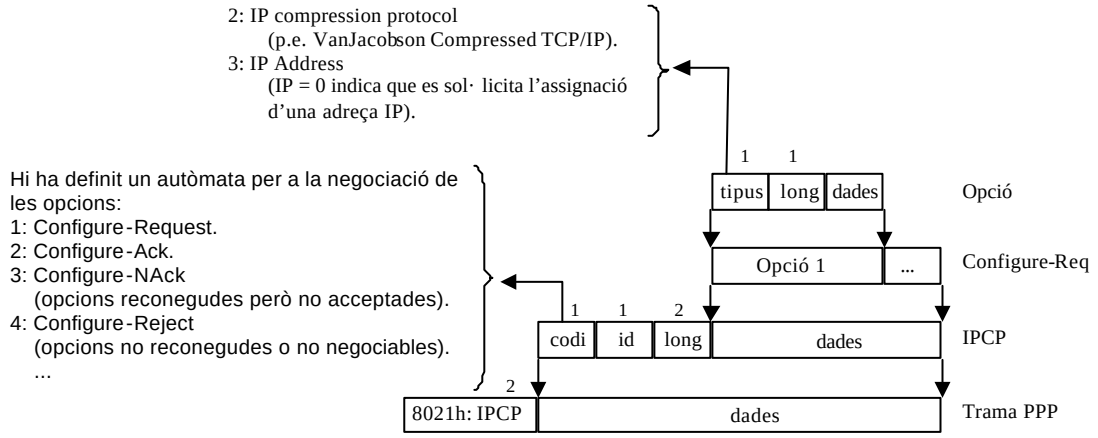


figura 7: Format i encapsulat d'una trama PPP amb informació IPCP.

3 Descripció de la comanda pppd

La comanda `pppd` permet l'establiment d'un enllaç PPP en una màquina linux. Si la connexió es fa a través del port sèrie mitjançant un modem, `pppd` sol fer-se servir conjuntament amb el programa `chat` per a comunicar-se amb el modem i establir la connexió telefònica abans de que `pppd` entri en acció. Si la connexió es fa a través del port seria mitjançant un cable *null modem*, aleshores el programa `chat` no és necessari.

La invocació de la comanda `pppd` és:

`pppd [ tty_name ] [ options ]`

on **`tty_name`** és la interfície (COM1 i COM2 es corresponen amb linux a `/dev/ttyS0` i `/dev/ttyS1` respectivament) i **`options`** son les opcions. Les opcions es llegeixen primer del fitxer `/etc/ppp/options` (si existeix) i després de la línia de comandes. Algunes de les opcions possibles son:

`defaultroute`

Add a default route to the system routing tables, using the peer as the gateway, when IPCP negotiation is successfully completed. This entry is removed when the PPP connection is broken. This option is privileged if the nodefaultroute option has been specified.

`asyncmap <map>`

Set the async character map to <map>. This map describes which control characters cannot be suc cessfully received over the serial line. Pppd will ask the peer to send these characters as a 2-byte escape sequence. The argument is a 32 bit hex number with each bit representing a character to escape. Bit 0 (00000001) represents the character 0x00; bit 31 (80000000) represents the character 0x1f or ^_. If multiple asyncmap options are given, the values are ORed together. If no asyncmap option is given, no async character map will be negotiated for the receive direction; the peer should then escape all control characters. To escape transmitted characters, use the escape option.

`mru n`

Set the MRU [Maximum Receive Unit] value to n. Pppd will ask the peer to send packets of no more than n bytes. The minimum MRU value is 128. The default MRU value is 1500. A value of 296 is recommended for slow links (40 bytes for TCP/IP header + 256 bytes of data). (Note that for IPv6 MRU must be at least 1280)

`mtu n`

Set the MTU [Maximum Transmit Unit] value to n. Unless the peer requests a smaller value via MRU negotiation, pppd will request that the kernel networking code send data packets of no more than n bytes through the PPP network interface. (Note that for IPv6 MTU must be at least 1280)

`passive`

Enables the "passive" option in the LCP. With this option, pppd will attempt to initiate a connection; if no reply is received from the peer, pppd will then just wait passively for a valid LCP packet from the peer, instead of exiting, as it would without this option.

`<local_IP_address>:<remote_IP_address>`

Set the local and/or remote interface IP addresses. Either one may be omitted. The IP addresses can be specified with a host name or in decimal dot notation (e.g. 150.234.56.78). The default local address is the (first) IP address of the system (unless the noipdefault option is given). The remote address will be obtained from the peer if not specified in any option. Thus, in simple cases, this option is not required. If a local and/or remote IP address is specified with this option, pppd will not accept a different value from the peer in the IPCP negotiation, unless the ipcp-accept-local and/or ipcp-accept-remote options are given, respectively.

`bsdcomp nr,nt`

Request that the peer compress packets that it sends, using the BSD-Compress scheme, with a maximum code size of nr bits, and agree to compress packets sent to the peer with a maximum code size of nt bits. If nt is not specified, it defaults to the value given for nr. Values in the range 9 to 15 may be used for nr and nt; larger values give better compression but consume more kernel memory for compression dictionaries. Alternatively, a value of 0 for nr or nt disables compression in the

corresponding direction. Use nobsdcomp or bsdcomp 0 to disable BSD-Compress compression entirely.

debug

Enables connection debugging facilities. If this option is given, pppd will log the contents of all control packets sent or received in a readable form. The packets are logged through syslog with facility daemon and level debug. This information can be directed to a file by setting up /etc/syslog.conf appropriately (see syslog.conf(5)).

deflate nr,nt

Request that the peer compress packets that it sends, using the Deflate scheme, with a maximum window size of 2**nr bytes, and agree to compress packets sent to the peer with a maximum window size of 2**nt bytes. If nt is not specified, it defaults to the value given for nr. Values in the range 8 to 15 may be used for nr and nt; larger values give better compression but consume more kernel memory for compression dictionaries. Alternatively, a value of 0 for nr or nt disables compression in the corresponding direction. Use nodeflate or deflate 0 to disable Deflate compression entirely. (Note: pppd requests Deflate compression in preference to BSD-Compress if the peer can do either.)

local

Don't use the modem control lines. With this option, pppd will ignore the state of the CD (Carrier Detect) signal from the modem and will not change the state of the DTR (Data Terminal Ready) signal.

netmask n

Set the interface netmask to n, a 32 bit netmask in "decimal dot" notation (e.g. 255.255.255.0). If this option is given, the value specified is ORed with the default netmask. The default netmask is chosen based on the negotiated remote IP address; it is the appropriate network mask for the class of the remote IP address, ORed with the netmasks for any non point-to-point network interfaces in the system which are on the same network. (Note: on some platforms, pppd will always use 255.255.255.255 for the netmask, if that is the only appropriate value for a point-to-point interface.)

noaccomp

Disable Address/Control compression in both directions (send and receive).

noauth

Do not require the peer to authenticate itself. This option is privileged.

nobsdcomp

Disables BSD-Compress compression; pppd will not request or agree to compress packets using the BSD-Compress scheme.

noccp

Disable CCP (Compression Control Protocol) negotiation. This option should only be required if the peer is buggy and gets confused by requests from pppd for CCP negotiation.

nocrtscts

Disable hardware flow control (i.e. RTS/CTS) on the serial port. If neither the crtscts nor the nocrtscts nor the cdtrcts nor the nodtrcts option is given, the hardware flow control setting for the serial port is left unchanged.

nodeflate

Disables Deflate compression; pppd will not request or agree to compress packets using the Deflate scheme.

noipdefault

Disables the default behaviour when no local IP address is specified, which is to determine (if possible) the local IP address from the hostname. With this option, the peer will have to supply the local IP address during IPCP negotiation (unless it specified explicitly on the command line or in an options file).

nopcomp

Disable protocol field compression negotiation in both the receive and the transmit direction.

novj

Disable Van Jacobson style TCP/IP header compression in both the transmit and the receive direction.

proxyarp

Add an entry to this system's ARP [Address Resolution Protocol] table with the IP address of the peer and the Ethernet address of this system. This will have the effect of making the peer appear to other systems to be on the local ethernet.

record filename

Specifies that pppd should record all characters sent and received to a file named filename. This file is opened in append mode, using the user's user-ID and permissions. This option is implemented using a pseudo-tty and a process to transfer characters between the pseudo-tty and the real serial device, so it will increase the latency and CPU overhead of transferring data over the ppp interface. The characters are stored in a tagged format with timestamps, which can be displayed in readable form using the pppdump(8) program.

silent

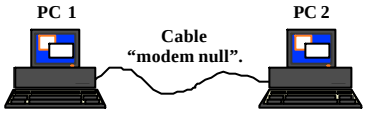
With this option, pppd will not transmit LCP packets to initiate a connection until a valid LCP packet is received from the peer (as for the `passive' option with ancient versions of pppd).

usepeerdns

Ask the peer for up to 2 DNS server addresses. The addresses supplied by the peer (if any) are passed to the /etc/ppp/ip-up script in the environment variables DNS1 and DNS2. In addition, pppd will create an /etc/ppp/resolv.conf file containing one or two nameserver lines with the address(es) supplied by the peer.

4 Realització de la pràctica

La pràctica consisteix en experimentar alguns aspectes d'una connexió PPP entre dos PCs connectats amb un cable "modem null" a través del port sèrie com mostra la figura. Primer de tot hi ha que botar els dos PCs amb els discs de trinux. Seguiu les instruccions que es donen a continuació. Les tasques a realitzar estan indicades amb un marc.



4.1 Configuració del port

Abans d'executar pppd és necessari que la connexió a través del cable *modem null* funcioni correctament. Per veure si hi ha connectivitat es pot fer el següent test: executar en PC1

```
cat < /dev/ttyS0
```

i en PC2

```
echo hola > /dev/ttyS0
```

Si hi ha connectivitat en PC1 hauria d'aparèixer el missatge `hola`. Per a veure la configuració, i per a configurar el port sèrie es fan servir les comandes `setserial` i `stty`. `setserial` dóna informació relativa al *driver*, i `stty` dóna informació relativa a la configuració de la línia. Per exemple, si executem:

```
bash# setserial –a /dev/ttyS0
/dev/ttyS0, Line 0, UART: unknown, Port: 0x03f8, IRQ: 4
      Baud_base: 115200, close_delay: 50, divisor: 0
      closing_wait: 3000
      Flags: spd_normal skip_test
```

Amb el missatge "UART: unknown" podem veure que la UART no està correctament configurada. Per a saber el tipus de UART de que disposem podem mirant el fitxer `/var/log/messages`, en alguna indicació generada en el *boot* del sistema referent a `ttyS0`. Si la UART és la 16550A (actualment és la més habitual) hem d'executar:

```
bash# setserial /dev/ttyS0 uart 16550A
```

Amb `stty` podem veure quina és la configuració de la línia executant:

```
bash# stty –aF /dev/ttyS0
```

Perquè hi hagi connectivitat amb el cable *modem null*, els dos ports han de tenir la mateixa velocitat de transmissió (valor de `speed` al executar `stty`), i paritat (`parodd`). Podem saber el significat dels indicadors de la comanda `stty` executant `stty --help`.

- Comproveu que teniu connectivitat i configureu si cal la UART.

4.2 Establiment de la connexió PPP

Per a una primera connexió PPP executeu en PC1 i PC2 respectivament les comandes:

```
pppd /dev/ttyS0 debug local 10.0.0.1:10.0.0.2 record pc1.bin noccp -detach | tee pc1.asc &
pppd /dev/ttyS0 debug local noipdefault record pc2.bin noccp -detach | tee pc2.asc &
```

Podeu comprovar si la connexió s'ha establert mirant amb la comanda `ifconfig` si la interfície `ppp0` ha quedat configurada.

- Amb la informació de la secció 3 identifica les opcions de les comandes anteriors.
- Investiga amb les comandes `ifconfig` i `route` les modificacions que l'establiment del enllaç PPP ha introduït en la interfícies `ppp` i la taula d'encaminament.
- Interpreteu els missatges que apareixen en la pantalla durant la connexió. Fixeu-vos que aquests missatges també queden emmagatzemats en els fitxers `pc1.asc` i `pc2.asc`. Deduïu a partir d'aquests missatges quines opcions negocia el protocol LCP i IPCP.

4.3 Anàlisi de les trames

Per a llegir el contingut dels fitxers `pc1.bin` i `pc2.bin` generats amb les comandes `pppd` anteriors, ha de usar-se la comanda `pppdump`. Amb la opció `–hp`, `pppdump` fa el volcat en hexadecimal i agrupa en una mateixa línia els caràcters que pertanyen a una mateixa trama.

- Executa `pppdump –hp pc1.bin` en PC1. Identifica tots els camps i el seu significat del primer paquet LCP enviat. Ajuda't amb el fitxer `pc1.asc` i la descripció dels camps que hi ha en la figura 5 i la figura 6.

Opcional:

- Executa `tcpdump –i ppp0` per a capturar les trames intercanviades al fer `telnet` al port `daytime`. Mira amb `pppdump` el camp de protocol de les trames ppp intercanviades per comprovar que estan comprimides.
- Tanca la connexió anterior executant `killall pppd` en els dos PCs i torna a establir l'enllaç PPP eliminant l'opció `noccp` en un dels dos PCs.
- Investiga els paquets intercanviats pel protocol CCP per veure com un dels PCs proposa un algorisme de compressió i l'altra el rebutja.
- Executa `tcpdump –i ppp0` per a capturar les trames intercanviades al fer un telnet al port `daytime`. Mira amb `pppdump` el camp de protocol de les trames PPP intercanviades per comprovar que estan comprimides.