simple isdn4linux PPP FAQ .. to be continued .. not 'debugged'

Q01: what's pppd, ipppd, syncPPP, asyncPPP ??
Q02: error message "this system lacks PPP support"
Q03: strange information using 'ifconfig'
Q04: MPPP?? What's that and how can I use it ...
Q05: I tried MPPP but it doesn't work
Q06: can I use asynchronous PPP encapsulation with network devices
Q07: A SunISDN machine can't connect to my i4l system
Q08: I wanna talk to several machines, which need different configs
Q09: Starting the ipppd, I get only error messages from i4l
Q10: I wanna use dynamic IP address assignment
Q11: I can't connect. How can I check where the problem is.
Q12: How can I reduce login delay?

Q01: pppd, ipppd, syncPPP, asyncPPP .. what is that ?
what should I use?
A: The pppd is for asynchronous PPP .. asynchronous means here, the framing is character based. (e.g when using ttyI* or tty* devices)

The ipppd handles PPP packets coming in HDLC frames (bit based protocol) ... The PPP driver in isdn4linux pushes all IP packets direct to the network layer and all PPP protocol frames to the /dev/ippp* device. So, the ipppd is a simple external network protocol handler.

If you login into a remote machine using the /dev/ttyI* devices and then enable PPP on the remote terminal server -> use the 'old' pppd

If your remote side immediately starts to send frames ... you probably connect to a syncPPP machine .. use the network device part of isdn4linux with the 'syncppp' encapsulation and make sure, that the ipppd is running and connected to at least one /dev/ippp*. Check the isdn4linux manual on how to configure a network device.

--

Q02: when I start the ipppd .. I only get the
error message "this system lacks PPP support"
A: check that at least the device 'ippp0' exists.
(you can check this e.g with the program 'ifconfig')
The ipppd NEEDS this device under THIS name ..
If this device doesn't exists, use:

```
     isdnctrl addif ippp0
     isdnctrl encap ippp0 syncppp
     ... (see isdn4linux doc for more) ...
```

A: Maybe you have compiled the ipppd with another

```
   kernel source tree than the kernel you currently
   run ...

--

Q03: when I list the netdevices with ifconfig I see, that
   my ISDN interface has a HWaddr and IRQ=0 and Base
   address = 0
A: The device is a fake ethernet device .. ignore IRQ and baseaddr
   You need the HWaddr only for ethernet encapsulation.

--

Q04: MPPP?? What's that and how can I use it ...

A: MPPP or MP or MPP (Warning: MP is also an
   acronym for 'Multi Processor') stands for
   Multi Point to Point and means bundling
   of several channels to one logical stream.
   To enable MPPP negotiation you must call the
   ipppd with the '+mp' option.
   You must also configure a slave device for
   every additional channel. (see the i4l manual
   for more)
   To use channel bundling you must first activate
   the 'master' or initial call. Now you can add
   the slave channels with the command:
       isdnctrl addlink <device>
   e.g:
       isdnctrl addlink ippp0
   This is different from other encapsulations of
   isdn4linux! With syncPPP, there is no automatic
   activation of slave devices.

--

Q05: I tried MPPP but it doesn't work .. the ipppd
   writes in the debug log something like:
   .. rcvd [0][proto=0x3d] c0 00 00 00 80 fd 01 01 00 0a ...
   .. sent [0][LCP ProtRej id=0x2 00 3d c0 00 00 00 80 fd 01 ...

A: you forgot to compile MPPP/RFC1717 support into the
   ISDN Subsystem. Recompile with this option enabled.

--

Q06: can I use asynchronous PPP encapsulation
   over the network interface of isdn4linux ..

A: No .. that's not possible .. Use the standard
   PPP package over the /dev/ttyI* devices. You
   must not use the ipppd for this.

--

Q07: A SunISDN machine tries to connect my i4l system,
```

which doesn't work.
Checking the debug log I just saw garbage like:
!![... fill in the line ...]!!

A: The Sun tries to talk asynchronous PPP ... i4l can't understand this ... try to use the ttyI* devices with the standard PPP/pppd package

A: (from Alexanter Strauss:)
!![... fill in mail]!!

--

Q08: I wanna talk to remote machines, which need a different configuration. The only way I found to do this is to kill the ipppd and start a new one with another config to connect to the second machine.

A: you must bind a network interface explicitly to an ippp device, where you can connect a (for this interface) individually configured ipppd.

--

Q09: When I start the ipppd I only get error messages from the i4l driver ..

A: When starting, the ipppd calls functions which may trigger a network packet. (e.g gethostbyname()). Without the ipppd (at this moment, it is not fully started) we can't handle this network request. Try to configure hostnames necessary for the ipppd in your local /etc/hosts file or in a way, that your system can resolve it without using an isdn/ippp network-interface.

--

Q10: I wanna use dynamic IP address assignment ... How must I configure the network device.

A: At least you must have a route which forwards a packet to the ippp network-interface to trigger the dial-on-demand.
A default route to the ippp-interface will work.
Now you must choose a dummy IP address for your interface.
If for some reason you can't set the default route to the ippp interface, you may take any address of the subnet from which you expect your dynamic IP number and set a 'network route' for this subnet to the ippp interface.
To allow overriding of the dummy address you must call the ipppd with the 'ipcp-accept-local' option.

A: You must know, how the ipppd gets the addresses it wanna configure. If you don't give any option, the ipppd tries to negotiate the local host address! With the option 'noipdefault' it requests an address from the remote machine. With 'useifip' it gets the addresses from the net interface. Or you set the address on the option line with the <a.b.c.d:e.f.g.h> option. Note: the IP address of the remote machine must be configured locally or the remote machine must send it in an IPCP request. If your side doesn't know the IP address after negotiation, it closes the connection! You must allow overriding of address with the 'ipcp-accept-*' options, if you have set your own or the remote address explicitly.

A: Maybe you try these options .. e.g:

```
/sbin/ipppd :$REMOTE noipdefault /dev/ippp0
```

where REMOTE must be the address of the remote machine (the machine, which gives you your address)

--

Q11: I can't connect. How can I check where the problem is.

A: A good help log is the debug output from the ipppd... Check whether you can find there:

- only a few LCP-conf-req SENT messages (less then 10) and then a Term-REQ:
  -> check whether your ISDN card is well configured it seems, that your machine doesn't dial (IRQ,IO,Proto, etc problems) Configure your ISDN card to print debug messages and check the /dev/isdnctrl output next time. There you can see, whether there is activity on the card/line.
- there are at least a few RECV messages in the log:
  -> fine: your card is dialing and your remote machine tries to talk with you. Maybe only a missing authentication. Check your ipppd configuration again.
- the ipppd exits for some reason:
  -> not good ... check /var/adm/syslog and /var/adm/daemon. Could be a bug in the ipppd.

--

Q12: How can I reduce login delay?

A: Log a login session ('debug' log) and check which options your remote side rejects. Next time configure your ipppd to not negotiate these options. Another 'side effect' is, that this increases redundancy. (e.g your remote side is buggy and rejects options in a wrong way).