

Package PPPOE

Version 4.0.0-r60785-testing

Frank Meyer
email: frank@fli4l.de

Das fli4l-Team
email: team@fli4l.de

October 14, 2022

Contents

1. Documentation For Package PPPOE	3
1.1. DSL - DSL over PPPoE, Fritz!DSL and PPTP	3
1.1.1. General Configuration Variables	3
1.1.2. OPT_PPPOE - DSL over PPPoE	6
1.1.3. OPT_FRITZDSL - DSL via Fritz!Card DSL	8
1.1.4. OPT_PPTP - DSL over PPTP in Austria/the Netherlands	9
1.1.5. OPT_POESTATUS - PPPoE-Status-Monitor On fli4l-Console	10
A. Appendix For Package PPPOE	11
A.1. DSL - PPPD and Active Filter	11
List of Figures	12
List of Tables	13
Index	14

1. Documentation For Package PPPOE

1.1. DSL - DSL over PPPoE, Fritz!DSL and PPTP

fli4l supports DSL in three different variants:

- PPPoE (external DSL-modems connected over ethernet using pppoe)
- PPTP (external DSL-modems connected over ethernet using pptp)
- Fritz!DSL (DSL over DSL-adapters manufactured by AVM)

You can choose only one of these options, simultaneous operation isn't possible yet. The configuration for all variants is similar, so the general parameters are described at first and then the special options for the individual variants will be discussed. DSL-access is handled by imond as a circuit. Therefore it is necessary to activate imond by setting (see `OPT_IMOND` (Page ??)) to 'yes'.

1.1.1. General Configuration Variables

The packages all use the same configuration variables, they differ only by the package name prefixes. As an example: in all packages the user name is required. The variable is named `PPPOE_USER`, `PPTP_USER` or `FRITZDSL_USER` depending on the package. The variables are described indicating the missing prefix by an asterisk. In concrete examples PPPOE is assumed but these are valid with any other prefix too.

***_NAME** Set a name for the circuit - max. 15 digits long. This will be shown in the imon client imonc. Spaces (blanks) are not allowed.

Example: `PPPOE_NAME='DSL'`

***_USEPEERDNS** This specifies whether the address of the name servers given by the internet service provider at dial-in will be added to the configuration file of the local name server for the duration of the online connection or not.

It only makes sense to use this option for internet access circuits. Almost all providers support these type of transfer by now.

After the name server IP addresses have been transferred, name servers in `DNS_FORWARDERS` will be used as forwarders. Afterwards the local nameserver is forced to reread its configuration. Previously resolved names will not be lost from the name server cache.

This option has the advantage of always using the nearest name servers (given the provider transmits the correct IP addresses) thus accelerating the name resolution process.

In case of failure of a provider's DNS server in most cases the incorrect DNS server addresses are rapidly corrected by the provider.

1. Documentation For Package PPPOE

Nevertheless it is necessary to specify a valid name server in `DNS_FORWARDERS`. Otherwise the first name request can not be resolved correctly. In addition the original configuration of the local nameservers will be restored when terminating the connection.

Default-setting: `*_USEPEERDNS='yes'`

***_DEBUG** If you want pppd to be more verbose set `*_DEBUG` to 'yes'. In this case pppd will write additional informations to syslog.

Attention: In order to get this to work syslogd has to be activated by setting `OPT_SYSLOGD` to 'yes' in base.txt.

***_USER, *_PASS** Provide user ID and password for the provider used. `*_USER` is used for the user id `*_PASS` is the password.

Attention: For T-Online-access consider this:

Username `AAAAAAAAAAAAATTTTTT#MMMM` is put together from a 12 digit 'Anschlußkennung', T-Online-number and 'Mitbenutzernummer'. Behind the T-Online-number follows a '#' (only if its length is shorter than 12 digits).

If this does not work in rare cases put another '#' between 'Anschlußkennung' and T-Online-number.

User ID for T-Online has to have an additional '@t-online.de' at the end!

Example:

```
PPPOE_USER='111111111111222222#0001@t-online.de'
```

Infos on user ID's for other providers are found in the FAQ:

- http://extern.fli4l.de/fli4l_faqengine/faq.php?list=category&catnr=3&prog=1

***_HUP_TIMEOUT** Specify the time in seconds after which the connection will be terminated when no more traffic is detected over the DSL line. A timeout of '0' or 'never' stands for no timeout. Using 'never' the router immediately reconnects after a disconnection. Changing dialmodes is not possible then - it has to stay 'auto'. 'Never' is currently only available for PPPOE and FritzDSL.

***_CHARGEINT** Charge-interval: Time interval in seconds which will be used for calculating online costs.

Most providers calculate their charges per minute. In this case put in '60'. For providers that charge per second set `*_CHARGEINT='1'`.

Unfortunately for DSL the timing is not fully utilized, as it is the case for ISDN. In our case a hangup will be triggered after the time specified in `*_HUP_TIMEOUT`.

Hence `*_Chargeint` is only significant for the calculation of charges.

***_TIMES** This times determine when to enable the circuit and how much it will cost. This makes it possible to have different default routes at different times for a circuit (Least Cost Routing). The imond daemon controls route assignment then.

Composition of variables:

1. Documentation For Package PPPOE

```
PPPOE_TIMES='times-1-info [times-2-info] ...'
```

Each field times-?-info consists of 4 parts divided by colons (':').

1. Field: W1-W2

Weekday-period, i.e. Mo-Fr or Sa-Su aso. English and german notations are both valid. For a single weekday write W1-W1 (i.e. Su-Su).

2. Field: hh-hh

Hours, i.e. 09-18 or 18-09 too. 18-09 is equivalent to 18-24 plus 00-09. 00-24 means the complete day.

3. Field: Charge

Currency-values as costs per minute, i.e. 0.032 for 3.2 Cent per minute. They are used to calculate the actual costs incurred, taking into account the cycle time. They will be displayed in the imon client.

4. Field: LC-Default-Route

Content can be Y or N, which means:

Y: The specified time range will be used as default route for LC-routing.

N: The specified time range is only used for calculating costs but it won't be used for automatic LC-routing.

Example (read as one long line):

```
PPPOE_TIMES='Mo-Fr:09-18:0.049:N
             Mo-Fr:18-09:0.044:Y
             Sa-Su:00-24:0.039:Y'
```

Important: *Times used in *_TIMES have to cover the whole week. If that is not the case a valid configuration can't be build.*

If the time ranges off all LC-default-route-circuits ("Y") together don't contain the complete week there will be no default route in these gaps. There will be no internet access possible at these times!

One more simple example:

```
PPPOE_TIMES='Mo-Su:00-24:0.0:Y'
```

for those using a flatrate.

One last comment to LC-routing: *holidays are treated as Sundays.*

***_FILTER** fli4l hangs up automatically if there is no traffic over the pppoe interface in the time specified in hangup timeout. Unfortunately also traffic from the outside counts here, for example connection attempts by a P2P client such as eDonkey. Since you will be contacted almost permanently from outside nowadays, it can happen that fli4l never terminates the DSL connection.

Option `*_FILTER` comes to help here. Setting it to yes will only consider traffic that is generated from your own machine and external traffic will be completely ignored. Since incoming traffic usually means that the router or computers behind it respond by i.e rejecting requests additionally some outgoing packets are ignored. Read here how this exactly works:

- <http://www.fli4l.de/hilfe/howtos/basteleien/hangup-problem-loesen/> and
- <http://web.archive.org/web/20061107225118/http://www.linux-bayreuth.de/dcforum/DCForumID2/46.html>.

A more detailed description of the expression and its usage is to be found in the appendix but is only interesting if you want to make changes.

***_FILTER_EXPR** Filter to use if `*_FILTER` is set to 'yes'.

***_MTU *_MRU** These variables are optional and can be completely omitted.

With these optional variables the so-called **MTU** (maximum transmission unit) and the **MRU** (maximum receive unit) can be set. By default MTU and MRU are set to 1492. This setting should only be changed in special cases! These variables don't exist for `OPT_PPTP`.

***_NF_MSS** For some providers these effects can occur:

- Webbrowser gets a connection but is unresponsive after that,
- small mails can be sent but big mails can't,
- ssh works, scp hangs after initial connecting.

To work around this problems fli4l manipulates the MTU as a default. In some cases this is not enough so fli4l explicitly permits setting of the MSS (message segment size) to a value given by the provider. If the provider does not give any values 1412 is a good start to try. If a MTU is given by the provider, subtract 40 Byte here ($mss = mtu - 40$). These variables are optional and can be completely omitted. These variable doesn't exist for `OPT_PPTP`.

1.1.2. OPT_PPPOE - DSL over PPPoE

For communication via a DSL connection usually the PPPoE package is necessary because the provider does not provide a full blown router, but only a DSL modem. Between the fli4l router and the modem the PPP protocol is used, but in this case specifically over Ethernet.

One or two ethernet cards can be used in fli4l:

- only one card with IP for the LAN and PPP towards the DSL-modem
- two cards: one with IP for the LAN, the other for PPP towards the DSL-modem

The better choice is to use two ethernet cards. This way both protocols - IP and PPPoE - are clearly separated from each other.

But the method with one ethernet cards works as well. In this case the DSL-modem has to be connected to the network hub like all other clients. The maximum transfer speed can be slightly affected this way.

1. Documentation For Package PPPOE

If experiencing communication problems between the modem and the network card you can try to slow down the speed of the NIC, eventually even switching it to half-duplex mode. All PCI cards but only a few PCI-ISA network cards can be configured to run in various speed modes. Either use `ethtool` from the package `advanced_networking` or create a DOS boot media and include the native configuration tool there. Start `fli4l` with this media and execute the card's native tool to choose and save the slower operation mode to the card. The configuration tool usually is available on the driver media or can be downloaded from the website of the card manufacturer. You may also find it in a search in the wiki:

- <https://ssl.networks.org/wiki/display/f/Netzwerkkarten>

If using two network cards you should use the first for the LAN and the second for the connection to the DSL modem.

Only the first card does need an IP address.

This means:

```
IP_NET_N='1'           # Only *one* card with IP-address!
IP_NET_1xxx='...'      # Usual parameters
```

For `PPPOE_ETH` set the second ethernet card to `'eth1'` and define `*no*` `IP_NET_2-xxx`-variables.

OPT_PPPOE activates support for PPPoE. Default setting: `OPT_PPPOE='no'`.

PPPOE_ETH Name of the ethernet interface

```
'eth0'  first ethernet card
'eth1'  second ethernet card
...     ...
```

Default setting: `PPPOE_ETH='eth1'`

PPPOE_TYPE *PPPOE* stands for transmission of PPP-packets over ethernet lines. Data to be transmitted is transformed to ppp-packets in a first step and then in a second step wrapped in pppoe-packets to be transmitted over ethernet to the DSL-modem. The second step can be done by the `pppoe-daemon` or by the kernel. `PPPOE_TYPE` defines the way of pppoe packet generation.

Values	Description
async	Packets are generated by the <code>pppoe-daemon</code> ; asynchronous communication between <i>pppd</i> and <i>pppoed</i> .
sync	Packets are generated by the <code>pppoe-daemon</code> ; synchronous communication between <i>pppd</i> and <i>pppoed</i> . This way communication is more efficient and thus leads to lower processor load.
in_kernel	Packets are generated by the linux kernel. This way communication with a second daemon is omitted saving a lot of in-memory copying thus leading to even lower processor load.

Table 1.1.: Ways of generating pppoe packets

Somebody did a comparison between the various variants on a Fujitsu Siemens PCD-H, P75 the results are shown in table 1.2 ¹.

fli4l	NIC	Bandwidth (downstream)	CPU-load
2.0.8	rtl8029 + rtl8139	310 kB/s	100%
2.0.8	2x 3Com Etherlink III	305 kB/s	100%
2.0.8	SMC + 3Com Etherlink III	300 kB/s	100%
2.1.7	SMC + 3Com Etherlink III	375 kB/s	40%

Table 1.2.: Bandwidth und CPU-load with pppoe

PPPOE_HUP_TIMEOUT Using PPPoE-type in `_kernel` and dialmode auto, timeout can be set to 'never'. The router then no longer hangs up and after a forced disconnection by the provider dials again automatically. Subsequent changing of the dialmode is not possible anymore.

1.1.3. OPT_FRITZDSL - DSL via Fritz!Card DSL

Internet connection by Fritz! Card DSL is activated here. A Fritz! Card DSL by AVM is used for the internet connection. Since the drivers for these cards are not subject to the GPL it is not possible to provide them with the DSL package. It is essential to download these drivers before from <http://www.fli4l.de/download/stabile-version/avm-treiber/> and to extract them into the fli4l directory.

Circuit-support for Fritz!Card DSL was realised with friendly help from Stefan Uterhardt (email: zer0@onlinehome.de).

OPT_FRITZDSL activates support for Fritz!DSL. Default setting: `OPT_FRITZDSL='no'`.

FRITZDSL_TYPE Several Fritz!-cards exist for a DSL connection. The card used will be specified by `FRITZDSL_TYPE`, have a look at table 1.3 for enumerating the supported cards.

Card Type	Usage
fcdsl	Fritz!Card DSL
fcdsl2	Fritz!Card DSLv2
fcdslsl	Fritz!Card DSL SL
fcdslusb	Fritz!Card DSL USB
fcdslslusb	Fritz!Card DSL SL USB
fcdslusb2	Fritz!Card DSL USBv2

Table 1.3.: Fritz-Cards

Default setting:

```
FRITZDSL_TYPE='fcdsl'
```

¹Values were taken from a posting on spline.fli4l and have not been evaluated further. Message ID of the article: <caf9fk\$ala\$1@bla.spline.inf.fu-berlin.de>.

FRITZDSL_PROVIDER With this option the type of the remote station is set. Possible options are:

U-R2, ECI, Siemens, Netcologne, oldArcor, Switzerland, Belgium, Austria1, Austria2, Austria3, Austria4

In Germany almost always UR-2 is used. Siemens and ECI are only used with very old ports.

For Switzerland and Belgium, the options are self-explanatory and in Austria you have to try what works for you.

If anyone has a better description for the options in Austria please tell.

Default setting:

```
FRITZDSL_PROVIDER='U-R2'
```

1.1.4. OPT_PPTP - DSL over PPTP in Austria/the Netherlands

In Austria (and other european countries) PPTP-protocol is used instead of PPPoE. A separate ethernet card is connected to a PPTP-Modem in this case.

As of version 2.0 connecting via PPTP is realised as a Circuit - with the friendly help of Rudolf Hämmerle (email: rudolf.haemmerle@aon.at).

PPTP uses two cards. The first card should be used for connecting the LAN and the second for connecting to the DSL-modem.

Only the first card can have an IP-address.

This means:

```
IP_NET_N='1'                # Only *one* card with IP-address!
IP_NET_1xxx='...'           # the usual parameters
```

PPTP_ETH is set to 'eth1' for the second ethernet card and *no* IP_NET_2-xxx-variables are defined.

OPT_PPTP activates support for PPTP. Default setting: OPT_PPTP='no'.

PPTP_ETH Name of the ethernet interfaces

```
'eth0'  first ethernet card
'eth1'  second ethernet card
...     ...
```

Default setting: PPTP_ETH='eth1'

PPTP_MODEM_TYPE There are several PPTP modem types to realise a pptp connection. Define PPTP_MODEM_TYPE to define the modem type. Possible types are shown in table 1.4.

Default setting:

```
PPTP_MODEM_TYPE='telekom'
```

Modem Type	Usage
telekom	Austria (Telekom Austria)
xdsl	Austria (Inode xDSL@home)
mxstream	the Netherlands, Danmark

Table 1.4.: PPTP Modem Types

Inode xDSL@home

Support for Inode xDSL@home was implemented using information found on Inode's support pages².

At the moment there are sometimes problems with renewing of leases for the interface (the IP for the interface is delivered by dhcp and has to be renewed in regular periods). Hanging up and reconnecting via imonc does not work well by now. Help by providing patches or as a tester is highly appreciated.

With xsdl two further options exist for pptp:

PPTP_CLIENT_REORDER_TO The pptp-client which is used for xsdl under certain circumstances must rearrange and buffer packets. It usually waits 0.3 seconds for a packet. By setting this variable you can vary the timeout between 0.00 (no buffer) and 10.00. Times always must be provided with two decimales.

PPTP_CLIENT_LOGLEVEL Here you can define how much debug output will be produced by the pptp-client. Values are 0 (little), 1 (default) und 2 (much).

1.1.5. OPT_POESTATUS - PPPoE-Status-Monitor On fli4l-Console

PPPoE-Status-Monitor for DSL Connections was developed by Thorsten Pohlmann.

With setting OPT_POESTATUS='yes' dsl status can be watched on the third fli4l console at any time. Switch to the third console by pressing ALT-F3 and back to the first console with ALT-F1.

²See http://www6.inode.at/support/internetzugang/xdsl_home/konfiguration_ethernet_linux.html

A. Appendix For Package PPPOE

A.1. DSL - PPPD and Active Filter

fi4l uses the expression:

```
'outbound and not icmp[0] != 8 and not tcp[13] & 4 != 0'
```

and accomplishes that generally only packets sent from the local network to the internet keep the connection open, with a few exceptions:

- *TCP-RST*: Answers to rejected connection from outside do not reset the timeout,
- *ICMP*: ICMP messages sent do not reset the timeout unless an echo request is sent.

This expression is converted by the PPPD into a packet-filter usable by the kernel. In this example it looks like this:

```
#
# Expression: outbound and not icmp[0] != 8 and not tcp[13] & 4 != 0
#
(000) ldb      [0]
(001) jeq      #0x0          jt 17   jf 2
(002) ldh      [2]
(003) jeq      #0x21          jt 4    jf 18
(004) ldb      [13]
(005) jeq      #0x1          jt 6     jf 11
(006) ldh      [10]
(007) jset     #0x1fff        jt 18   jf 8
(008) ldx      4*([4]&0xf)
(009) ldb      [x + 4]
(010) jeq      #0x8          jt 18   jf 17
(011) jeq      #0x6          jt 12   jf 18
(012) ldh      [10]
(013) jset     #0x1fff        jt 18   jf 14
(014) ldx      4*([4]&0xf)
(015) ldb      [x + 17]
(016) jset     #0x4          jt 17   jf 18
(017) ret      #0
(018) ret      #4
```

List of Figures

List of Tables

1.1. Ways of generating pppoe packets	7
1.2. Bandwidth und CPU-load with pppoe	8
1.3. Fritz-Cards	8
1.4. PPTP Modem Types	10

Index

FRITZDSL_CHARGEINT, [4](#)
FRITZDSL_DEBUG, [4](#)
FRITZDSL_FILTER, [5](#)
FRITZDSL_FILTER_EXPR, [6](#)
FRITZDSL_HUP_TIMEOUT, [4](#)
FRITZDSL_MRU, [6](#)
FRITZDSL_MTU, [6](#)
FRITZDSL_NAME, [3](#)
FRITZDSL_NF_MSS, [6](#)
FRITZDSL_PASS, [4](#)
FRITZDSL_PROVIDER, [9](#)
FRITZDSL_TIMES, [4](#)
FRITZDSL_TYPE, [8](#)
FRITZDSL_USEPEERDNS, [3](#)
FRITZDSL_USER, [4](#)

OPT_FRITZDSL, [8](#)
OPT_POESTATUS, [10](#)
OPT_PPPOE, [7](#)
OPT_PPTP, [9](#)

PPPOE_CHARGEINT, [4](#)
PPPOE_DEBUG, [4](#)
PPPOE_ETH, [7](#)
PPPOE_FILTER, [5](#)
PPPOE_FILTER_EXPR, [6](#)
PPPOE_HUP_TIMEOUT, [4](#), [8](#)
PPPOE_MRU, [6](#)
PPPOE_MTU, [6](#)
PPPOE_NAME, [3](#)
PPPOE_NF_MSS, [6](#)
PPPOE_PASS, [4](#)
PPPOE_TIMES, [4](#)
PPPOE_TYPE, [7](#)
PPPOE_USEPEERDNS, [3](#)
PPPOE_USER, [4](#)
PPTP_CHARGEINT, [4](#)
PPTP_CLIENT_LOGLEVEL, [10](#)

PPTP_CLIENT_REORDER_TO, [10](#)
PPTP_DEBUG, [4](#)
PPTP_ETH, [9](#)
PPTP_FILTER, [5](#)
PPTP_FILTER_EXPR, [6](#)
PPTP_HUP_TIMEOUT, [4](#)
PPTP_MODEM_TYPE, [9](#)
PPTP_NAME, [3](#)
PPTP_PASS, [4](#)
PPTP_TIMES, [4](#)
PPTP_USEPEERDNS, [3](#)
PPTP_USER, [4](#)