

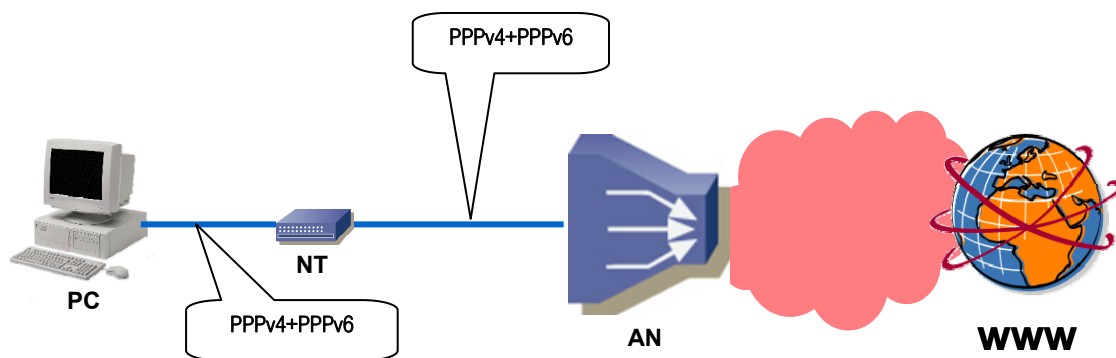
Settings for the operating system of the computer initiating the IPv6 network connection in case of direct connection (without the use of home networking devices)

The IPv6 connection in each case is placed by initiating another IPv6 based PPPoE session. The login name initiating the IPv6 connection is derived from the login for the traditional internet (IPv4), by adding @ipv6.t-online.hu instead of @t-online.hu postfix. The related password remains unchanged, identical to the one used to log on to the traditional internet service.

For example, if your username for traditional internet access is [username@t-online.hu](#), then upon setting up your new IPv6 connection, your login will be [username@ipv6.t-online.hu](#). The password does not change, but since you have to enter this again when making IPv6 settings, you have to remember your own password in order to make such settings.

To access the IPv6 test service, within the operating system you are using, you have to set up a new network connection, in addition to the already operational IPv4 PPPoE connection, which would start the new IPv6 based PPPoE connection.

The following figure describes how the devices and how they are connected to each other:



IMPORTANT!

The Microsoft Windows operating systems currently are not capable of handle more than one parallel PPPoE connections therefore they cannot be used during the pilot without installing a home networking device (router or HGW).

Linux systems can handle parallel/concurrent PPPoE network connections. If you want to activate IPv6 internet connection on Ubuntu Linux, please, follow the steps in the **Ubuntu Linux** chapter. In the **Other Unix/Linux distributions** chapter you can find general information based on which the configuration steps required for various distributions can be identified.

Ubuntu Linux

In case of Ubuntu Linux operating system, please go through the following steps in order to set up the IPv6 PPPoE dialer.

IMPORTANT!

The settings of the `/etc/ppp/options` file used by the `pppoeconf` script built into Ubuntu Linux are not the same in case of IPv4 and IPv6 connections, therefore separate dialers need to be created under different names for both connections. A comfortable solution is to use a ppp dialer that handles every PPP configuration in the same file.

It may be useful to first save all existing ppp settings (e.g. dsl provider) under a different filename (e.g. `ipv4-dsl-provider`) and continue with further steps only afterwards.

1. For the dialer to be working properly, certain lines of the `/etc/ppp/options` file should be modified by deleting or inserting the comment sign (`#`) at the appropriate positions.

Please, open a Terminal window (Menu>Applications>Accessories>Terminal) and type in the following command:

```
ubuntu@ubuntu: ~$ sudo gedit /etc/ppp/options
```

Now the editor opens the file in a new window. The lines to be modified are as follows: (IMPORTANT: this screen only shows the modified and not the unchanged lines! Those must not be deleted!):

```
# /etc/ppp/options
ipv6 ,

# 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).
noipdefault

# 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 does without this option).
#passive

# 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 old versions of pppd).
#silent

# Don't request or allow negotiation of any options for LCP and IPCP
# (use default values).
#-all

# Disable Address/Control compression negotiation (use default, i.e.
# address/control field disabled).
#-ac

# Disable IP address negotiation (with this option, the remote IP
# address must be specified with an option on the command line or in
# an options file).
-ip

# Disable IPCP negotiation and IP communication. This option should
# only be required if the peer is buggy and gets confused by requests
# from pppd for IPCP negotiation.
noip

# If this option is given, pppd will send an LCP echo-request frame to the
# peer every n seconds. Normally the peer should respond to the echo-request
# by sending an echo-reply. This option can be used with the
# lcp-echo-failure option to detect that the peer is no longer connected.
lcp-echo-interval 30
```

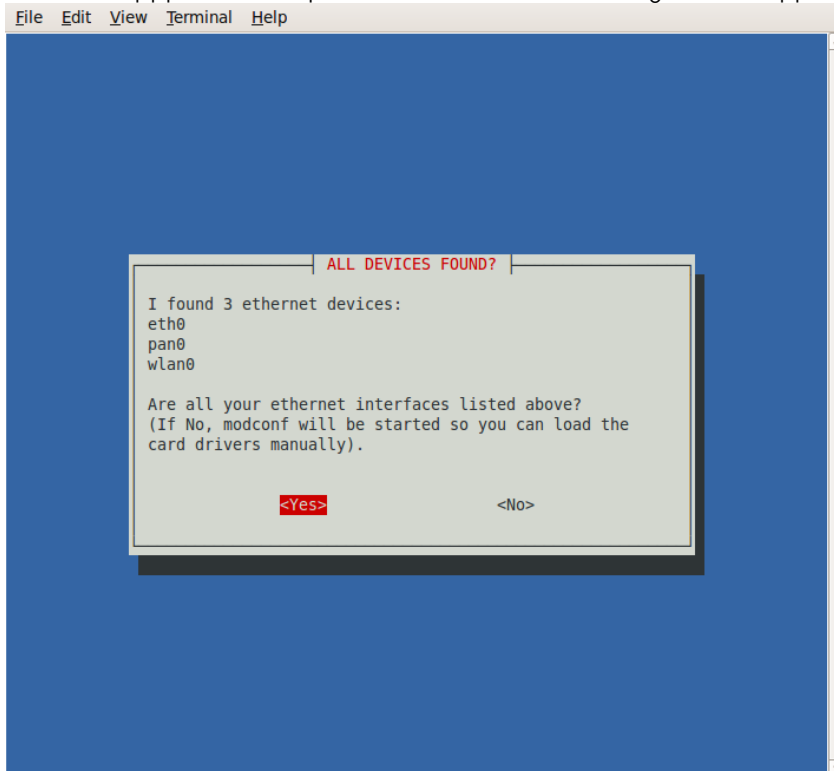
```
# If this option is given, pppd will presume the peer to be dead if n
# LCP echo-requests are sent without receiving a valid LCP echo-reply.
# If this happens, pppd will terminate the connection. Use of this
# option requires a non-zero value for the lcp-echo-interval parameter.
# This option can be used to enable pppd to terminate after the physical
# connection has been broken (e.g., the modem has hung up) in
# situations where no hardware modem control lines are available.
lcp-echo-failure 4
```

After making the modifications, please save the file under the same name and close gedit program.

2. After saving, returning to the Terminal window, please type in the following command: *"sudo pppoeconf"*

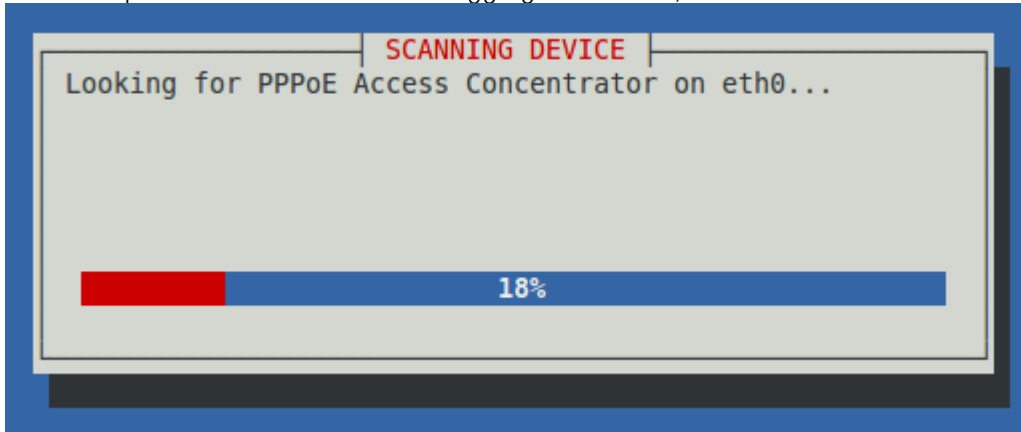
```
ubuntu@ubuntu: ~$ sudo pppoeconf
```

3. Now the pppoeconf script will launch and the following window appears:

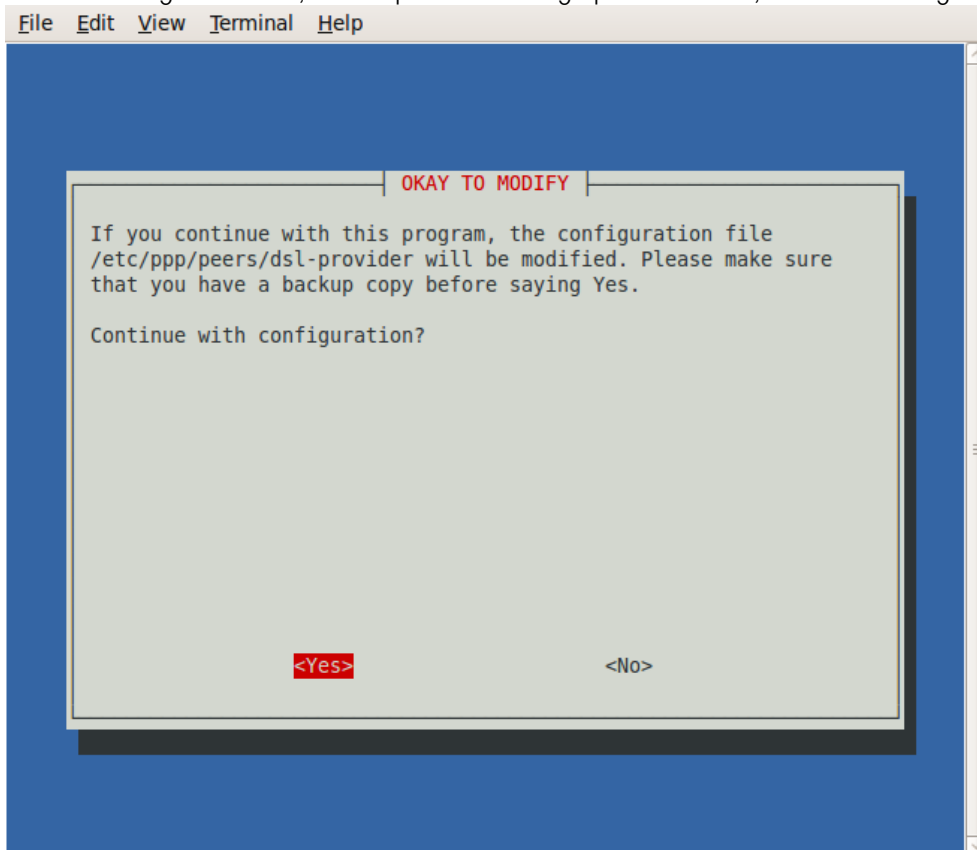


If the required device is on the list (in case of fixed connection this would be one of the lines starting with "eth"), please, select "<Yes>" and press Enter.

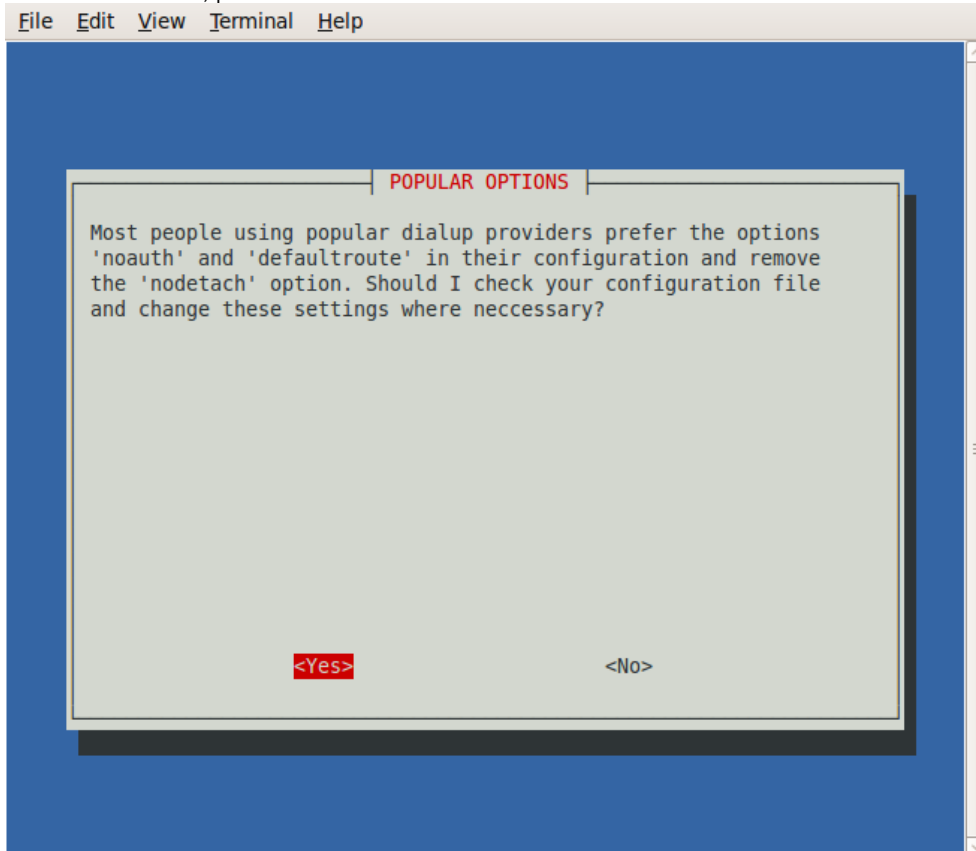
4. The script will now look for a remote aggregation device, it can connect to:



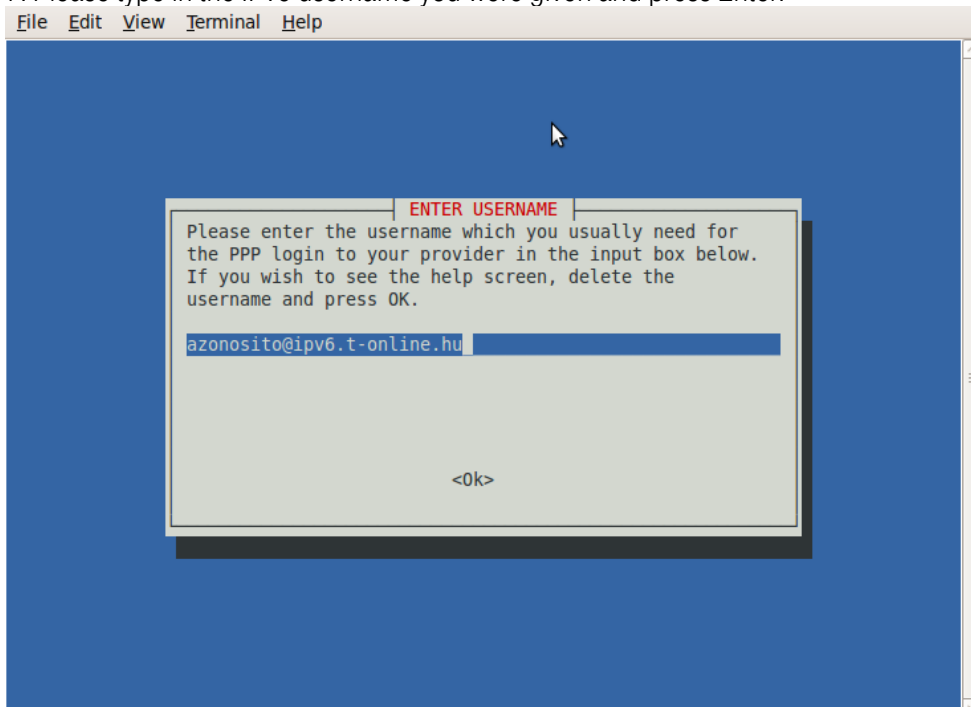
5. After finding the device, the script offers setting options. Please, select <Yes> again.



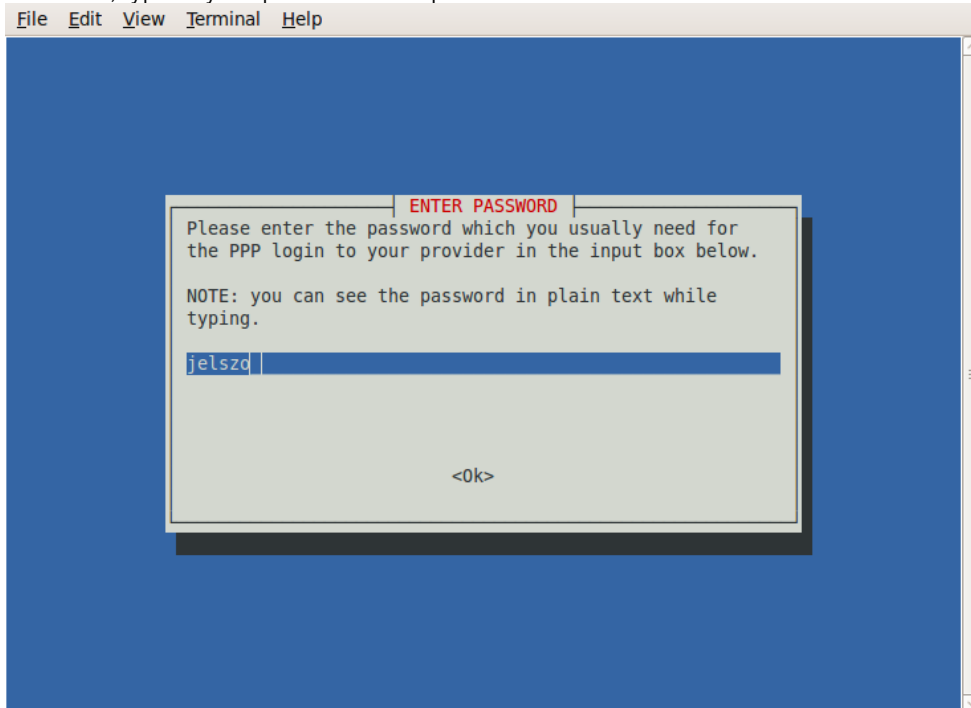
6. In this window, please select `<Yes>` as well.



7. Please type in the IPv6 username you were given and press Enter.



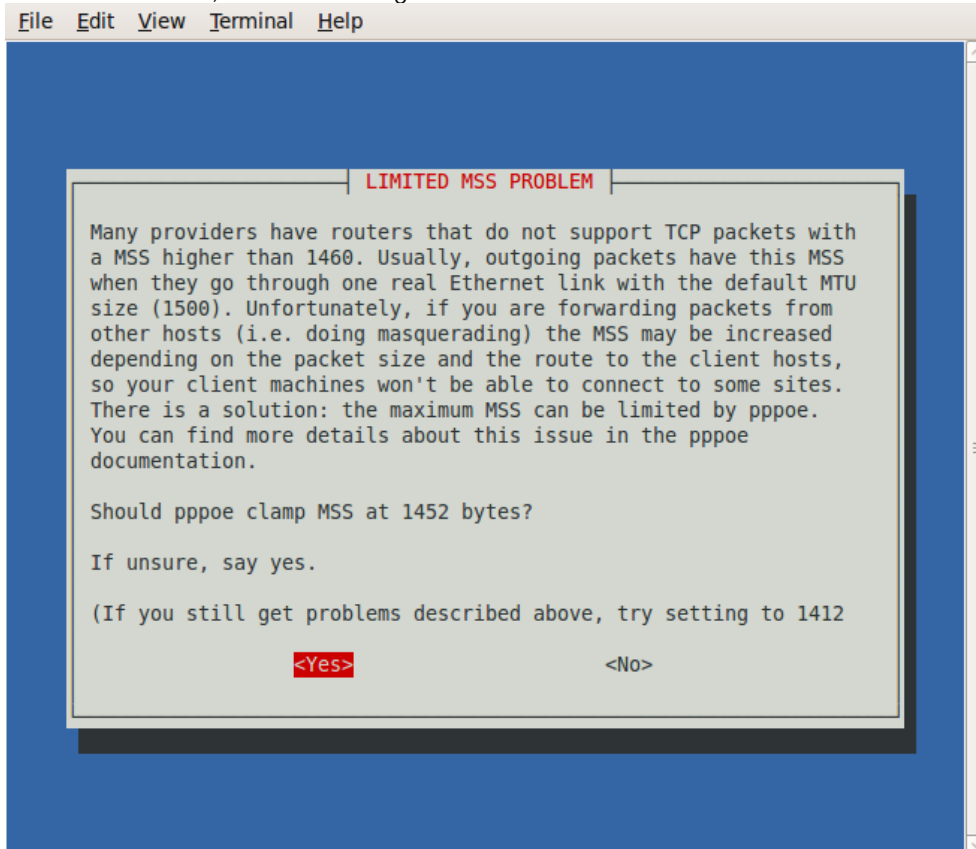
8. Please, type in your password and press Enter.



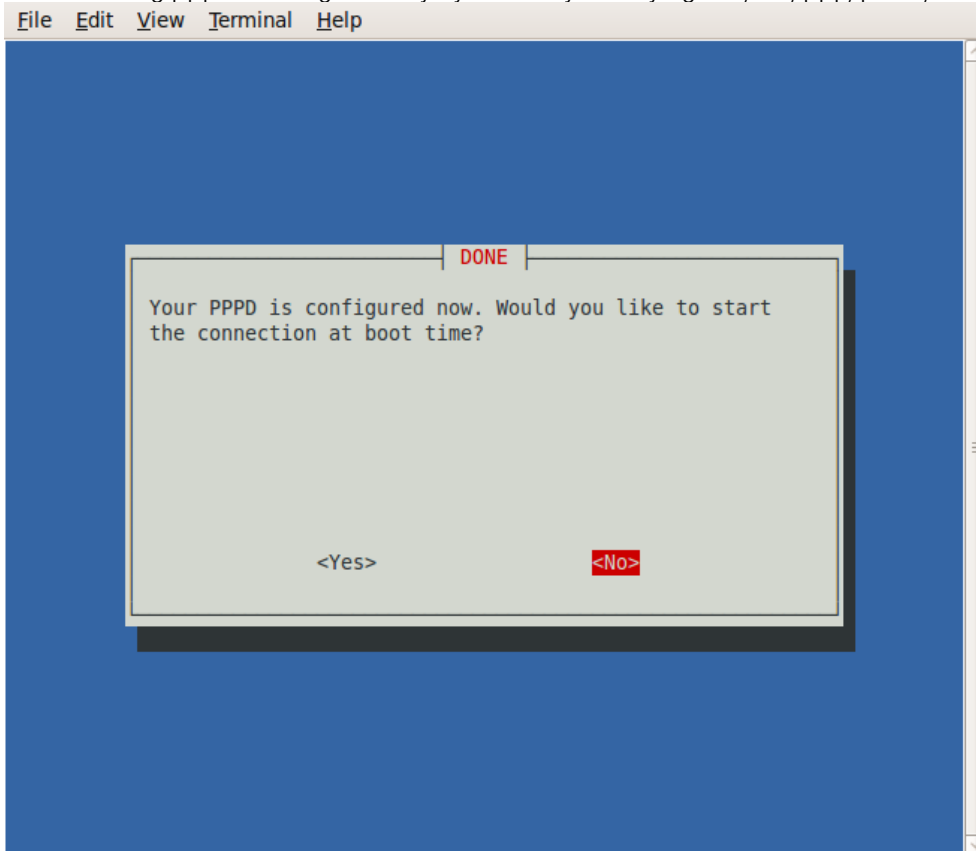
9. In this window, select <Yes> again.



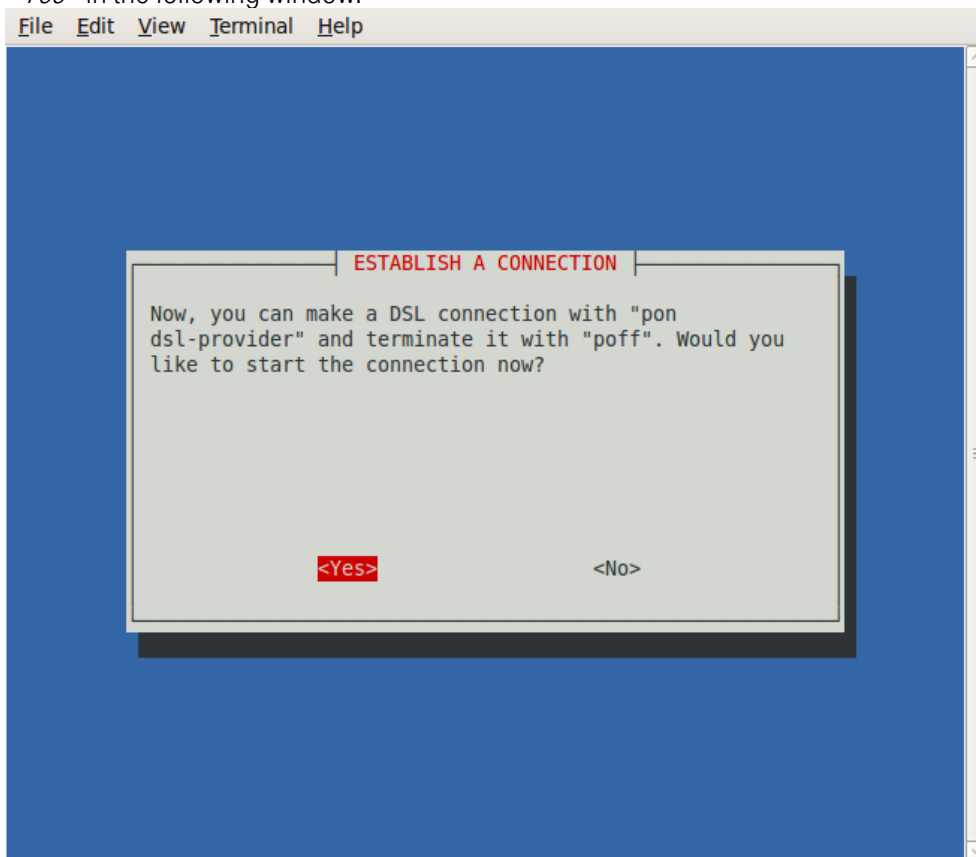
10. In this window, select `<Yes>` again



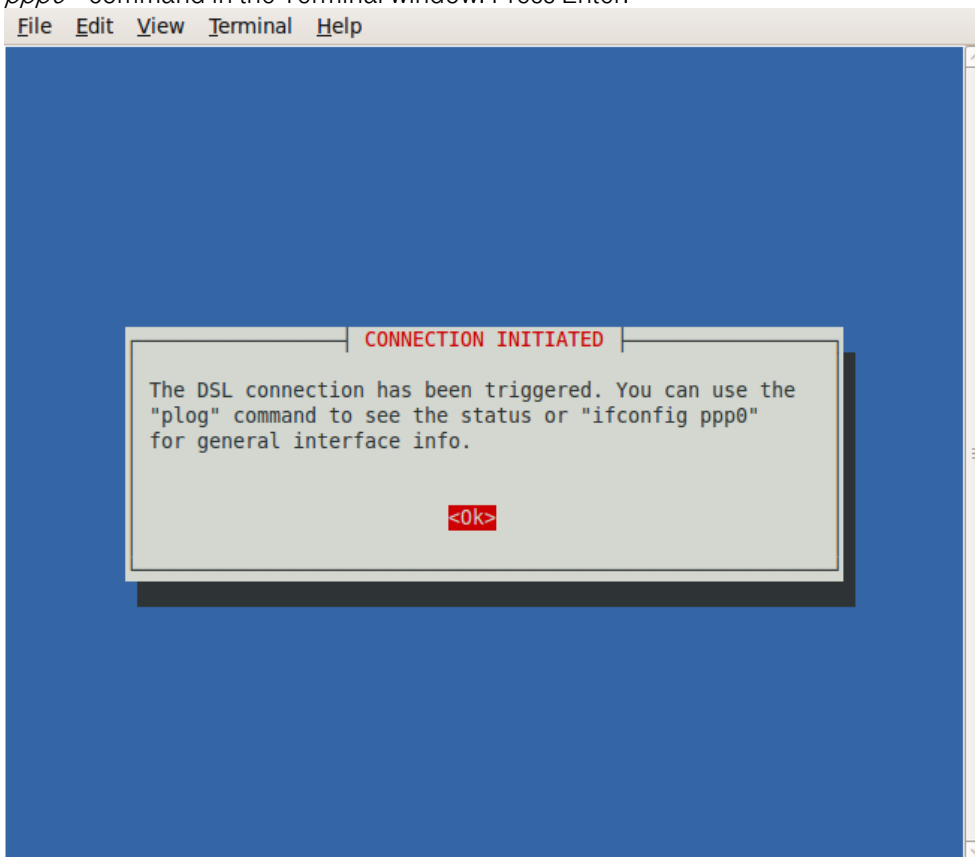
11. In this window you can freely decide about the following question: Would you like the computer to automatically start the IPv6 connection each time it boots? If yes, select *<Yes>*, if no, select *<No>*. You are advised to select the latter for greater controllability. (Naturally settings can be modified at any time, by either running `pppoeconf` again or by manually modifying the `/etc/ppp/peers/dsl-provider` file.)



12. The PPP dialer settings are now complete. If you would like to start the connection, please, select *<Yes>* in the following window.



13. You can check whether the connection was successful by entering the `sudo plog` or `sudo ifconfig ppp0` command in the Terminal window. Press Enter.



Should you want to start the connection again (if you selected `<No>` in step 10, or the connection was broken for some reason), you can do it by running the `sudo pon dsl-provider` command. If you would like to break the connection, run the `sudo poff` command.

You are advised to rename the recently created `dsl-provider` file as well. (e.g. `ipv6-dsl-provider`), this way later on the 2 dialer commands can clearly be differentiated. (naturally, the already changed names should be used for dialing from now on.)

You can check the connection status with the “*sudo plog*” or “*sudo ifconfig ppp0*” command.

```
File Edit View Terminal Help
root@kassait-HP-laptop:~#
root@kassait-HP-laptop:~# plog
Oct 22 16:22:25 kassait-HP-laptop pppd[4163]: RP-PPPoE plugin version 3.8p compiled against pppd 2.4.5
Oct 22 16:22:25 kassait-HP-laptop pppd[4165]: pppd 2.4.5 started by root, uid 0
Oct 22 16:22:25 kassait-HP-laptop pppd[4165]: PPP session is 59391
Oct 22 16:22:25 kassait-HP-laptop pppd[4165]: Connected to 00:14:f1:77:6c:1a via interface eth0
Oct 22 16:22:25 kassait-HP-laptop pppd[4165]: Using interface ppp0
Oct 22 16:22:25 kassait-HP-laptop pppd[4165]: Connect: ppp0 <-> eth0
Oct 22 16:22:28 kassait-HP-laptop pppd[4165]: PAP authentication succeeded
Oct 22 16:22:28 kassait-HP-laptop pppd[4165]: peer from calling number 00:14:F1:77:6C:1A authorized
Oct 22 16:22:28 kassait-HP-laptop pppd[4165]: local LL address fe80::9c3a:b206:c4e0:9e86
Oct 22 16:22:28 kassait-HP-laptop pppd[4165]: remote LL address fe80::020e:d6ff:feb9:d81b
root@kassait-HP-laptop:~#
root@kassait-HP-laptop:~# ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet6 addr: 2001:4c48:100:164:9c3a:b206:c4e0:9e86/64 Scope:Global
          inet6 addr: fe80::9c3a:b206:c4e0:9e86/10 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1492 Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:740 (740.0 B)  TX bytes:76 (76.0 B)

root@kassait-HP-laptop:~#
root@kassait-HP-laptop:~# █
```

IMPORTANT!

After establishing the IPv6 connection, a precondition to the automatic request for IPv6 addresses of the DNS servers is the existence of an installed and operational DHCPv6 client. In lack of this, DNS resolution only works on IPv4 basis. For this, you need to follow the steps below.

Installation of and settings for the required packages

The packages required for IPv6 operation can be installed by typing the following command in the Terminal window:

```
ubuntu@ubuntu: ~$ sudo apt-get install wide-dhcpv6-client
```

This is the DHCPv6 client which allows for prefix delegation and host configuration via IPv6 protocol.

The proper operation of the DHCPv6 client requires the following settings:

```
ubuntu@ubuntu: ~$ sudo gedit/etc/wide-dhcpv6/dhcp6c.conf
```

After opening the editor please modify the file as follows, then save it.

```
# Default dhcp6c configuration: it assumes the address is autoconfigured using
# router advertisements.

interface ppp0{
    send ia-pd 0;
    request domain-name-servers, domain-name;
    send domain-name-servers, domain-name;
};

id-assoc pd {
    prefix-interface eth0{
```

```
#az sla-id értéke 0-255-ig tetszőleges decimális szám lehet
#a PD-nél kapott hexadecimális /56 címet ezzel egészítjük ki 64 bitre
#jelen esetben ez 4
sla-id 4;
sla-len 8;
};
```

ubuntu@ubuntu: ~\$ sudo dhcp6c -d -D -f ppp0

This command, which starts the dhcp6 client, should be entered at each boot time.

(The *-d -D -f* command-parameters should only be typed in if you would like to see what happens as a result of the command, however, the program runs even without these. In case of proper operation the state=REQUEST expression is shown. Upon longer operation the DHCP request is renewed from time to time, indicated by the expression RENEW. If you launched the command with command-parameters, please, open another Terminal window for further work.)

In case of proper operation a Global address appears on the eth0 device. Type in the following command:

ubuntu@ubuntu: ~\$ sudo ifconfig eth0

Now you should see something like this (the important part is highlighted):

```
eth0      Link encap:Ethernet  HWaddr 00:17:31:e6:77:99
          inet6 addr: 2001:4c48:100:164:9c3a:b206:c4e0:9e86/64 Scope:Global
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:20 Base address:0xdc00
```

Testing IPv6 operation

With the following commands you can check if the IPv6 protocol files really exist:

ubuntu@ubuntu: ~\$ sudo test -f /proc/net/ipv6_inet6 && echo "Running kernel is IPv6 ready"

Upon successful command run, the output is:

Running kernel is IPv6 ready

ubuntu@ubuntu: ~\$ sudo /sbin/ifconfig -? 2>& 1 | grep -qw 'inet6' && echo "utility 'ifconfig' is IPv6-ready"

Upon successful command run, the output is:

utility 'ifconfig' is IPv6-ready

Routing Table settings

Although Linux generally makes the right routing table settings, for safety's sake, please, run the following command (each time you restart our computer):

ubuntu@ubuntu: ~\$ sudo route add -A inet6 default dev ppp0

The existence of a successfully established connection can be checked with the “*sudo ifconfig*” command.

```

File Edit View Terminal Help
root@kassait-HP-laptop:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1a:4b:5d:e3:05
          inet addr:192.168.1.66  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::21a:4bff:fe5d:e305/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1906 (1.9 KB)  TX bytes:3525 (3.5 KB)
          Interrupt:18

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:319 (319.0 B)  TX bytes:319 (319.0 B)

ppp0     Link encap:Point-to-Point Protocol
          inet6 addr: 2001:4c48:100:164:d428:dc6a:9c9e:1f04/64 Scope:Global
          inet6 addr: fe80::d428:dc6a:9c9e:1f04/10 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1492  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:740 (740.0 B)  TX bytes:76 (76.0 B)

root@kassait-HP-laptop:~#

```

Automation of commands

If you do not want to type in all required commands at each boot time, you have to complete the following steps.

Please, enter the following command:

```
ubuntu@ubuntu: ~$ sudo gedit /etc/init.d/script3.sh
```

This command creates a script called script3.sh, into which you have to copy the following commands:

```

#!/bin/bash
echo -e "\n A Létező ppp kapcsolatok bontása!\n"
poff -a
sleep 3

echo -e "\n AZ IPV6-os ppp kapcsolat létrehozása!\n"
pon dsl-provider
sleep 8
ifconfig ppp0

echo -e "\n A routing tábla beállítás!\n"
route add -A inet6 default dev ppp0
sleep 2

echo -e "\n Ha a dhcp6c fut, újraindítjuk azt!\n"
kill `cat /var/run/dhcp6c.pid`
dhcp6c ppp0
sleep 2
ifconfig eth0
sleep 2

echo "A script lefutott"

```

You have to give the right authorization for the file to run:

```
ubuntu@ubuntu: ~$ chmod +x script3.sh
```

If later you want to start the script only manually, then at each boot time, please type in the following line:

```
ubuntu@ubuntu: ~$ sudo ./etc/init.d/script3.sh
```

If you want to automatically run the command at system boot time, please, enter the following command:

```
ubuntu@ubuntu: ~$ sudo update-rc.d script3.sh defaults 91
```

This function can be switched off at any time, if you do not need it:

```
ubuntu@ubuntu: ~$ sudo update-rc.d -f script3.sh remove
```

You can check the settings with the following command:

```
ubuntu@ubuntu: ~$ ls -l /etc/rc?.d/*script3.sh
```

1. Other Unix/Linux distributions

In case of other Unix/Linux distributions the required steps are almost identical to the ones for Ubuntu settings. The *pppoeconf* script is available for any version but it is worth using a ppp dialer that handles all PPP configurations in the same file. For example, this in case of FreeBSD looks like this:

```
# $FreeBSD: src/etc/ppp/ppp.conf,v 1.10 2004/11/19 17:12:56 obrien Exp $
#####
default:
set log Phase Chat LCP IPCP CCP tun command
ident user-ppp VERSION (built COMPILATIONDATE)

# Ensure that "device" references the correct serial port
# for your modem. (cuad0 = COM1, cuad1 = COM2)
#
set device /dev/cuad1

set speed 115200
set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
\\\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\t TIMEOUT 40 CONNECT"
set timeout 180 # 3 minute idle timer (the default)
enable dns # request DNS info (for resolv.conf)

#ipv6cp disable
papchap:
#
# edit the next three lines and replace the items in caps with
# the values which have been assigned by your ISP.
#

set phone PHONE_NUM
set authname USERNAME
set authkey PASSWORD

set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.0 0.0.0.0
add default HISADDR # Add a (sticky) default route

pppoe_serv:
set device PPPoE:em0:pppoe_serv # replace x11 with your Ethernet device
set authname ipv4.azonosito@t-online.hu
set authkey ipv4jelszo
set dial
# ipv6cp disable
set login
#add default HISADDR

#több pppoe kapcsolat esetén
pppoe_serv2:
set device PPPoE:em0:pppoe_serv2 # replace x11 with your Ethernet device
set authname azonosito@ipv6.t-online.hu
set authkey ipv6jelszo
set dial
# ipv6cp disable
set login
#add default HISADDR
```

In case of the above configuration you can start the IPv4 PPP connection by entering the *ppp -background pppoe_serv* command into the terminal window, then you can enter *ppp -background pppoe_serv2* which starts the IPv6 connection.

IMPORTANT!

After establishing the IPv6 connection, a precondition to the automatic request for the IPv6 addresses of the DNS servers is the existence of an installed and operational DHCPv6 client. In lack of this, DNS resolution only works on IPv4 basis. For this, you need to follow the steps below.

Installation and settings of the required packages

Please, install the package required for IPv6 operation with its dependencies on your computer:

wide-dhcpv6-client

This is the DHCPv6 client that allows for prefix delegation and host configuration via IPv6 protocol.

The proper operation of the DHCPv6 client requires the following settings to be made on the */etc/wide-dhcpv6/dhcp6c.conf* file.

After opening the editor, please, modify the file as follows, then save it.

```
# Default dhcp6c configuration: it assumes the address is autoconfigured using
# router advertisements.

interface ppp0{
    send ia-pd 0;
    request domain-name-servers, domain-name;
    send domain-name-servers, domain-name;
};

id-assoc pd {
    prefix-interface eth0{
        #az sla-id értéke 0-255-ig tetszőleges decimális szám lehet
        #a PD-nél kapott hexadecimális /56 címet ezzel egészítjük ki 64 bitre
        #jelen esetben ez 4
        sla-id 4;
        sla-len 8;
    };
};
```

This command, which starts the dhcp6 client, should be typed in at each boot time:

dhcp6c -d -D -f ppp0

(The *-d -D -f* command-parameters should only be typed in if you would like to see what happens as a result of the command, however, the program runs even without these. In case of proper operation the state=REQUEST expression is shown. Upon longer operation the DHCP request is renewed from time to time, indicated by the expression RENEW. If you launched the command with command-parameters, please, open another Terminal window for further work.)

In case of proper operation a Global address appears on the eth0 device. Type in the following command:

ifconfig eth0

Now you should see something like this (the important part is highlighted):

```
eth0      Link encap:Ethernet  Hwaddr 00:17:31:e6:77:99
          inet6 addr: 2001:4c48:100:164:9c3a:b206:c4e0:9e86/64 Scope:Global
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:20 Base address:0xdc00
```

Now the Unix/Linux computer settings are complete, the operating system is IPv6 capable.

If you do not want to type in all required commands at each boot time, their analogue automation is possible with Ubuntu Linux. For this, please, scroll back to the end of the previous chapter, to the “Automation of commands” section.