# pppstatus

# kill -9 phone bill

Despite cheap per minute deals and "capped" flat-rates with online time or download volume limits, it still makes sense to keep an eye on your connection data, and thus avoid surprises when the next phone bill arrives.

### BY MICHAEL RENNER AND PATRICIA JUNG

oes a low-cost flat-rate without an online time or traffic volume limit sound too good to be true? If you suspect that it is, pppstatus by Gabriel Montenegro can help you check out your connection data, by monitoring the traffic generated by modem or DSLbased Internet connections with time or volume restricted tariffs. Version 0.4.2 has matured noticeably. The software no longer simply shows the current statistics, but calculates the Internet charges that have already accrued.

A block graphic like the one shown in Figure 1 helps visualize the traffic volume. It would be hard to imagine better graphics in a console window, if you

want to keep the use of system resources to a minimum.

### First install...

Debian users are lucky, as a binary .deb format package is available. Simply log on as root and run the apt-get install pppstatus command, to download and install the package. Other distribution users will need to download the source code from http://pppstatus.sourceforge.

net/, and follow the usual steps:

tar xzf PPPStatus-₽ v0.4.2.tar.gz cd PPPStatus-v0.4.2 make

This assumes that you have the C compiler, the make-Tool and the ncurses library including the developer package (this is neurses-devel for SuSE). If the latter is missing, you will see an error message similar to the following:

pppstatus.c:34:20: curses.h: 2 File or directory not found

immediately after you call make. If make outputs a line with Compiled.. in it, you can assume that all the pre-requisites were fulfilled. In this case su to root and place the software on your system by typing *make install* .

# ... then Configure

Debian users will discover the required configuration files in the /etc/pppstatus directory, where pppstatus.cfg is the cen-

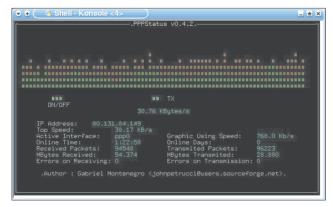


Figure 1: Data on the move

Table 1: pppstatus.cfg Settings		
Variable	possible values	Meaning
interface	ррро, ррр1, ррр2,	Network interface of modem
speed	14.4,28.8,33.6,56,64, 115.2,128,256,768,1540	Data throughput to the Internet pro- vider in Kbits/s
video	vga, mono	Display type
bytes_info	bytes, kbytes, mbytes	Transfer volume unit
user_email	Login name	user, whose mail account is to be monitored
costs_file	Path and filename, e.g./etc/pppstatus/costs	Filename of charge table



tral configuration file. But no matter what distribution you use, there is no avoiding a few customization steps to specify your network adapter and the maximum network throughput, that is the speed of your Internet connection. If you compiled the tool yourself, pppstatus will create this file in /var/log/ pppstatus.cfg when you launch the program for the first time.

You can also set the colors of the user interface to your personal preferences (Listing 1 shows an example). You can choose from black, white, red, green, yellow, blue and even magenta. Adding the light prefix will lighten the color.

If the configuration actually reflects

your computer hardware, the pppstatus command launch an empty overview on the console or in an xterm. If you have already connected to the Internet, the empty display will fill up with block graphics within a few seconds to indicate the network throughput.

### **Automatic**

If this initial test works out, you will probably want to launch the program automatically in future. If you tell the program to launch automatically when you start up your system, the graphic output will be sent to a virtual console. However, a few more customization steps are required to leverage the full potential of the program.

The Debian binary package includes the /etc/init.d/pppstatus file, which runs pppstatus

when you boot up your machine. You can use this file to specify the virtual console that pppstatus will output to. Most Linux systems have many virtual screens that can be accessed by pressing [Alt-F1] through [Alt-F12]. Any virtual console above F7 would be suitable for this purpose, as they are rarely used. You can change the TTY variable in the TTY = 12 line to any number between 7 and 11. 12, the last virtual console on most machines, is the default.

The Debian start script checks whether the /etc/pppstatus/pppstatus\_on \_boot file exists. If not, pppstatus will not launch, as you might have suspected. The best thing to do is to create the file - either by renaming the file created by the Debian installation, /etc/pppstatus/no\_pppstatus\_on\_boot, or by running the following command

```
touch /etc/pppstatus/⊋
pppstatus_on_boot
```

to create a new empty file. Having completed this step, you can use the following command

### /etc/init.d/pppstatus start

to simulate an automatic startup on booting. If you do not see an error message, press [Alt-Fx] to the console you specified as the output target, and ensure that everything is running to your liking.

If you compiled *pppstatus* on Red Hat, Slackware, or Debian, change back to the *PPPStatus-v0.4.2* source directory and setup the autostart function using the *runonboot*:

# Listing 1: Color definitions in /etc/pppstatus/pppstatus.cfg

```
ingoing = light green # incoming data
outgoing = yellow  # outgoing data
intersection = green # intersection of incoming and outgoing data
background = black  # background color
data = cyan  # font color for data values
border = white # border color
labels = white # label color
version = light white # color of program version display
power_led_on = light green  # status display with Internet
connection
power_led_off = red  # status display without Internet connection
```

### ./runonboot -d rh -t 9

will tell *R*ed *H*at to place output from *pppstatus* on virtual console number 9.

To automatically call the script on boot up, using the *start* argument, you still need to create a few links; Listing 2 does this for Red Hat. If you have a newer SuSE distribution replace /etc/rc.d/init.d/ with /etc/init.d/. If in doubt, the documentation for your distribution should tell you the directories you need.

## **Logging Online Costs**

pppstatus' integrated charge checker is not particularly flashy, but it is flexible enough to collate additional dial-up costs for Internet-by-call users. The data are stored in <code>/var/log/costs</code> (or in <code>/etc/pppstatus/costs</code> if you have the Debian binary). You can specify different connection charge per minute separately for weekdays, Saturday and Sunday.

If the overview has launched, any *ppp-status* instances launched subsequently

will be monitored, provided the *pppstatus.cfg* line is set to

```
costs_file = costs
```

(you may need to remove the hash sign # at the start of the line).

This last step provides you with an overview of the total transfer volume in bytes, at the same time monitoring the Internet-by-Call charges to help you avoid that dreaded phone bill shocker.

# Listing 3: Extract from costs file

```
# This file reflects Internet
connection charges per hour
monetary\_sign = US$
Currency
pulse_charge_time = 1
                            # Unit is
one minute
pulse_on_connect = yes # Dial-up
charge 'yes' or 'no'
on_connect_cost = 0.079 # Dial-up
charge cost
# wd (Work Days) i.e. Monday
through Friday
# sat = Saturday, sun = Sunday
00:00 \text{ wd} = 0.083
01:00 \text{ wd} = 0.083
02:00 \text{ wd} = 0.083
03:00 \text{ wd} = 0.083
04:00 \text{ wd} = 0.083
05:00 \text{ wd} = 0.083
06:00 \text{ wd} = 0.083
07:00 \text{ wd} = 0.083
08:00 \text{ wd} = 0.083
09:00 \text{ wd} = 0.079
10:00 \text{ wd} = 0.079
```

# Listing 2: Linking the Start-Stop Script

```
In -s /etc/rc.d/init.d/pppstatus /etc/rc.d/rc3.d/S75pppstatus
In -s /etc/rc.d/init.d/pppstatus /etc/rc.d/rc5.d/S75pppstatus
In -s /etc/rc.d/init.d/pppstatus /etc/rc.d/rc0.d/K20pppstatus
In -s /etc/rc.d/init.d/pppstatus /etc/rc.d/rc3.d/K20pppstatus
In -s /etc/rc.d/init.d/pppstatus /etc/rc.d/rc5.d/K20pppstatus
In -s /etc/rc.d/init.d/pppstatus /etc/rc.d/rc6.d/K20pppstatus
```

### **GLOSSARY**

**Ncurses:** A program library that allows simple graphic elements such as menus, to be displayed on the console.

**Virtual console:** Major Linux systems support multiple "monitors" where you can log on and run programs. Each of these "monitors",

or virtual consoles, displays a single screen. A combination of [Alt] with a function key toggles between screens. To switch from a GUI desktop to a text-based console, additionally press the [Ctrl] key.

[...]