# Sat−HOWTO

# Table of Contents

# Table of Contents

# Sat (Satellite Technology) HOWTO


## Roberto Arcomano berto@fatamorgana.com, Florindo Santoro flosan@hack−it.net



v1.7, 17 December 2001


---


*Sat technology is starting to become a great resource for Internet users, allowing high bandwidth in downloading and many other interesting services. This document wants to investigate "State of Art" of Sat connections in Linux environment, how to get them speeder and to share with many clients. You can found latest version of this document at http://www.fatamorgana.com/bertolinux and http://www.hack−it.net/How−To/Sat−HOWTO.html.*


---

## 1. Introduction



## 2. Background Knowledge



## 3. Technical Information



## 4. Requirements

## 5. Base setup



## 6. Configuration under Linux



## 7. Configuration under Windows



## 8. Appendix A − Notes



## 9. Appendix B − Well Known Sat ISP setting value



## 10. Links



---

# 1. Introduction

## 1.1 Introduction

This document wants to explain something about Satellite technology, how it works, what do you need, configuration and how to sharing it between several clients. Satellite connections are very different from terrestrial ones, they require more attention to setup and also some more care to maintain them stable (snow or strong rain could prevent you to have a good signal).

Feedback are welcome, don't hesitate to contact us: berto@fatamorgana.com and flosan@hack−it.net.

## 1.2 Copyright



## 1.3 Translations

If you want to translate this document you are free, you only have to:

1. Check that another version of it doesn't already exist at your local LDP
2. Maintain all 'Introduction' section (including 'Introduction', 'Copyright', 'Translations', 'Credits').

Thank you for your translation!

## 1.4 Credits

Thanks to Fatamorgana Computers for hardware equipment and experimental opportunity.

Thanks to Linux Documentation Project for publishing and uploading my document in a very quickly fashion.

Thanks to Pierre Guiral and Andrei Boros for their help.

---

# 2. Background Knowledge

## 2.1 What about satellite

In the last few years satellite began to be applied in Internet networking, mainly by medium−big ISPs and we have seen it diffused between users. Sat connections are a very different kind of networking than terrestrial ones, with different timing such as higher RTT (round trip time), but also with different bandwidth value, up to 2 Mbps or more.

## 2.2 How does it work?

We can imagine a path like this:

```
              |||||| S A T E L L I T E  ||||||||
                   /                       /|\
Downl             /                         |      Uploading
load             /                          |      from
to              /(4)                        |  (3)  server
client         /                            |
              /                             |
SatCard(parabolic antenna)                  |
             |                              |
           \|/                              |
       USER PC ----make request-----> SAT-SERVER <---retrieving---> INTERNET
                         (1)                                (2)
```

So first we make the request (1) (using our Internet connection) to the Sat−Server, after it will retrieve out info from Internet (2) and it will send it to Satellite (3); in the end we would receive data from the it (4) to our home using a parabolic antenna and a Sat Card.

Typically exist 2 kinds of request :

- Http
- Ftp

Both of them have a little request data and a much bigger answer size, so satellite works very well with it, but with a big time of answer, this is the biggest problem of satellite connection (consider a typical Sat distance, like 36.000 km, so you would have a time access of [ 36000 km / 300.000 km/s = 0.120 s = ] 120 ms you have to add (2 times, cause first ISPs server send it to, then you download it from) to classical Internet time access.

Recently ISP allows their clients to use also other kind of service, like

- Chat
- Email
- News

and many others.

There exist also services called "one−way", which consist in mail service , download on−demand (where you make a file reservation to be scheduled at some time) and site download: these services are offline, so you can access to them without modem (or other kind of) connection.

## 2.3 Other technologies

We have to report another satellite technology: the 2−way sat connection. It uses only the parabolic antenna to contact Internet in both 2 directions. Its cost is much higher that 1−way connection. We expect something from it for immediate future, for now it allows a bandwidth of 4 Mbit/s in download and 256Kbit/s in upload.

## 2.4 What is the max bandwidth?

It depends on many factors: ISP purpose, TCP window used, applications used by the client and the more important of all, "Internet congestion".

You can expect a max bandwidth of 1−4 Mbps and a average of about 10−30 KBytes/s, but I repeat, it depends on many factors.

Anyway some ISPs tell you they give you a "Max" bandwidth, while the average bandwidth could be very lower, due to intra−ISP congestion.

Some other ISPs guarantees you to have a "Minimal" bandwidth, which is more meaningful than "MAX", cause it is available all the time.

Please see Appendix A for more about get downloading performance better.

## 2.5 How much does it cost?

We have to distinguish between hardware costs and account costs, the first are known, while the second depend on which service you choose ("guaranteed"/"not guaranteed", what bandwidth).

### Hardware costs

To install our little satellite system we need:

1. Sat card
2. Parabolic antenna
3. one or more feed (converter)

Noticed that we need a digital converter to use Internet via satellite.

Sat card costs depends on brand you choose, about 200US$−300US$.

Parabolic antenna is about 50 US$.

Converter is about 50US$.

So we have about 3−400 US$ of hardware cost (maybe you still have to add installation cost!!).

If you want also receiver Crypt service:

1. CI is about 100 US$.
2. CAM is about 100 US$−250 US$.

### Account costs

Here costs depend on what ISPs give you access, what's peak bandwidth, if there is a guaranteed bandwidth (which is more important than peak one), what kind of service they give you and so on.

Also some ISP gives you free access in change of viewing an always foreground spot banner (you cannot iconize it!!): in this case you will pay when you'll go to buy something showed in banner!!

Typically account costs are about 100−150 US$ at year for "not guaranteed" services and 4−600 US$ or more for guaranteed ones (they guarantee you a minimal bandwidth you can use also under congestion moments, obviously intra Sat−ISP congestion!! When you go out to Internet nobody can guarantee you anything!!).

## 2.6 What are the satellites used?

When we speak about satellites we mean: Astra (19.2 degree SE), Hotbird (13.2 degree SE), new Europestar (45 degree SE), Eutelsat (8 degree SW), Astra (26E), ArabSat 3A (26E).

## 2.7 What ISPs give Sat Internet Access?

In Europe we know about 8 ISPs giving Sat access for Internet:

1. EuropeOnLine − EON
2. Netsystem
3. Starspeeder
4. Sat Node
5. Eliosat
6. Falcon Stream
7. SkyDSL
8. OpenSky
9. DirecPC but this use a proprietary sat card (this is a first Sat ISP )
10. http://www.xanticbroadband.com/

EON gives access for about 150 US$ at year without guaranteed bandwidth.

Netsystem offers its services at "null cost", you just have to see its banner.

Starspeeder gives access ???

Eliosat costs 350 US$ at year with a 128 Kbps guaranteed (minimal service, see web site for more);in addition it gives 2−way technology access for either receive and transmit.

SkyDSL gives access full−time and it costs about 15 US$ at month with 128Kbit/s bandwidth, but it allows a bigger bandwidth where you pay each Mb downloaded (you can select from 256Kbit/s up to 4Mbit/s), for more you can visit its web site.

IMPORTANT : before subscribing some satellite account, please verify "foot of Satellite" and diameter of parabola.

OpenSky started before summer 2001 in its "beta test": it allows you to try the sat service downloading at maximum 300 MB at month (free). To register you need to go at  OpenSky registration procedure (which is in italian language!).

## 2.8 What else?

With every kind of DVB card you can also receive TV digital channels (free channels only) and some cards have support for common interface to watch encrypted channel.

Follows the schema:

```
Smart−Card −> CAM −> Common Interface −> Sat Card (with support C.I.)
```

CAM Card (there are many standards used for decryption: SECA, IRDETO, VIACCESS and others) is the hardware allowing decryption (for TV, Radio and Data) while the Common Interface or C.I. (ETSI EN 50221) allow connection between CAM and Sat Card.

---

# 3. Technical Information

## 3.1 General working

We now try to understand how satellite connection works and at what conditions.

We can imagine a satellite link as a classical Wireless link, I mean a link between 2 systems which don't use a real cable to talk each other.

Wireless link is very different from Wired link cause we have some additional problems to solve, such as reachability, privacy problems and so on. Also there could be weather problems, particularly in snow or rain conditions.

Anyway, we have to consider the first principle behind Wireless communication: line of sight free, which is a MUST unless we are unable to talk. For more you can see the Wireless−HOWTO.

In sat connections we use a special kind of antenna, a parabolic one, that gives us a very high gain in RX, needed to receive satellite signal: in fact satellite has a geostational orbital at 36.000 km and the only kind of antenna we can use for receiving is just a parabolic one.

## 3.2 Frequencies

Frequency we receive is from 11GHz up to 12.7 GHz (from the satellite transponder, the transmitter sending us datas), a very high freq., but the feed (converter in the center of the parabola) converts it to, in output, 1−2 GHz so that we'll able to send signal to the receiver through the cable (up to 40m depending to cable loss).

## 3.3 DVB Receiver Card

```
1 GHz Signal --> |RX|--> |ADC| --> |Low Level Network| --> |O.S. TCP/IP Stack|--> Data
                 |_____________________________________|
                                DVB Card
```

Now we can imagine a classical RX at 1 GHz receiving analog signals from the Sat, converting it to digital signals and giving all to the low level network layer (ISO OSI 1,2): here, card firmware builds a 2 level packet (pretty like ethernet) to be sent to our PC with Linux, Windows, or other system, and in the end, we will only have to transform it to a TCP/IP packet.

## 3.4 Data Link Level DVB setting

Here we have to config some settings, directly to the DVB card:

1. Frequency working, we have to set the satellite frequency (as from magazines): it is between 11.8 GHz and 12.8 GHz for example 12640000 KHz
2. Symbol rate, measured in symbols per seconds, typically 22 MS/s (Mega Symbols per seconds)
3. Polarization, which is an antenna setting configurable in software: possible values are H (Horizontal) or V (Vertical).
4. PIDs. The PID is used for select a trasmission between many signal from same frequency.
5. Also we have to specify the bitfilter mask, which tells us how large is the destination MAC addresses group (bitfilter mask is similar to TCP/IP netmask, the only difference is that the bitfilter mask is 48 bit size while TCP/IP netmask is 32 bit size). Always Internet data is sent with Unicast PID, while Video always is sent using Multicast PID .
6. Calculate MAC address: here we specify what will be our MAC address (for our DVB card). You need to do this cause some ISP uses an algorithm (see Appendix A for more) which calculates your MAC address from your IP address, some other ISP uses your own MAC address.

## 3.5 TCP/IP working

As we said in 2.2 section, first we have to make a request using the modem interface (i.e. ppp0 or whatever we use to reach Internet), then the answer will return to our DVB interface (dvb0).

Modern O.S. allow us to receive packets from an input interface, different from the output interface from where we made the request: to do it we have to "disable" some packets flow control, such as type an

echo "0" > /proc/sys/net/ipv4/conf/dvb0/rp_filter (for Linux).

## 3.6 Authentication

It remains only one thing to complete our description: authentication method.

Some Sat ISPs use the so called "Proxy Authentication": when you used their proxy, you also need to give login and password to continue the request (you should have been subscribed some kind of account to use their sat service): once done, the ISP use your IP address to calculate your MAC address (see Appendix A for more), to which send the answer.

Some other ISPs require you make a VPN connection (using your login and password) first, then they will control your registration account (where they retrieve your MAC address) and will send data directly to (and only to) your card (your MAC address).

Anyway noticed that you can modify your dvb sat filter value to be able to receive packets destined to EVERY mac address (related to a single frequency).

## 3.7 Services I can use with a sat connection

Typically services you can have from sat connections depend on what authentication system is used by ISP:

- with classical "Proxy Authentication" you can only use HTTP and FTP service, while
- with VPN connection you have (ideally) no limit to services you can use, the only limit is decided by RTT (time access) to satellite, about 500−1000 ms average: so you can forget to use "Voice" or other "Real Time" services!! (see VoIP−HOWTO for more). Anyway you still can use: mail, chat, telnet, ping, dns and so on.

---

# 4. Requirements

Here we will see what we need to try a Sat system.

## 4.1 Knowledge requirement

For this trying you need some experience in internetworking under Linux (as from Net−HOWTO) and a very little of practical experience with parabolic antenna and sat systems (you should be able to pointing out your antenna, with right angles).

## 4.2 Hardware

We need:

1. Parabolic antenna, also taken from an analog system.
2. Digital converter to be placed in the middle of the antenna.
3. Receiver card DVB compatible.
4. a PC to connect to.

## 4.3 Software

Here you need a

1. driver for the DVB card working under your O.S. (Linux, Windows or other).
2. right setting to use it
3. external application in some cases (such as VPN−PPTP client for ISP like Netsystem or Starspeeder).

For the software under Linux you can found the Siemens DVB driver at Linux TV Project.

## 4.4 TV Software

There is also some Video software used to implement TV reception:

### Under Windows

1. MultiDec  MultiDec 6.6b (free with source code)
2. TPREdit  TPREdit (for Technotrend)
3. WinTV DVBs  Hauppauge or  Technotrend(for Siemens compatible card, Hauppauge, Technotrend, Technisat, etc... )
4. WinDVB2000  WinDVB2000 (about 19$).

### Under Linux

1. gVideo is an application included in Siemens driver
2. Vdr Video Recoder (good for watch TV)

---

# 5. Base setup

## 5.1 First: mounting

The first thing we have to do is to mount our parabolic antenna;

After we need to pointing it out (searching right degrees from some magazine): degrees are always intended from south to east or to west for horizontal one and from ground to satellite line for vertical one. Classical pointing tools is the compass.

## 5.2 Calibrating

How can we see if we are right oriented?

After decided a right range of angles, we have to adjust it measuring power level. For such a thing we can

1. Use for example an analog receiver (which is compatible with digital converter) and start to search right degree: when we see some imagine we'll get it. I suggest to use analog converter cause, maybe, you have one (from precedent installation) and cause it is more simple to calibrate it than digital one.
2. Use a power meter (about 20US$) with sequential lights: at full signal all lights turn up.

## 5.3 Double Feed

You can also install a double feed system (some vendors sell a complete kit with standard distance to receive, for example, Astra (19.2 SE) with Eutelsat (16 SE) or with HotBird (13 SE).

For mounting it you have to consider, in addition, that satellite is at the opposite side of the converter, like in figure:

```
           SAT1                 SAT2
              \                  /
               \                /
                \   C1   C2    /
         \    \    \ /    /     /
          \    \   / \   /     /
```

```
   \____\ /___\ /_____/

        Top View

 C1 receives from SAT2
 C2 receives from SAT1
```

Also, with 2 focus, you would use a diseq.

## 5.4 Data−link level adjusting

Once we got analog signal we have to adjust our receiver to right frequency, PID, speed rate and so on.

I report here an example of configuration, for EON (EuropeOnLine), transponder 114 on Astra satellite (19.2 SE)

Frequency: 12640 MHz

Polarization: V (Vertical)

Symbol Rate: 22000 KS/s

PIDs:

- Unicast: 512 (decimal), 0x200 (hexadecimal)
- Multicast: 785, 786, 1041 (decimal), 0x311, 0x312, 0x411 (hexadecimal), but you should notice that, in many cases, Multicast PIDs are taken in automatic.

We also need another info: what MAC address to assign to our DVB card.

Again, for EON you can see Appendix A to calculate MAC address from dynamic IP address.

Obviously you need login and password to use ISP service.

---

# 6. Configuration under Linux

In this section I will assume to use a Siemens compatible card, like an Hauppage WinTV DVB card, for such cards you can download drivers from  LinuxTV or  DVB−s PCI cards under Linux.

## 6.1 Drivers installation

Once downloaded drivers, you have to untar them to a directory, enter it and type "make" and "make insmod". To do this you need to have actual kernel sources under /usr/src/linux (unless, download them from http://www.kernel.org and recompile them).

After made "make insmod", your system should have DVB modules loaded. To unload them type simply "make rmmod".

## 6.2 Setting up /etc/dvbd.conf

/etc/dvbd.conf file is used to setup data−link parameters for your DVB card. Here main settings:

- "power", which indicates the power to LNB, 1=ON, 0=OFF. Usually you should set it to 1, unless you do some kind of cascade between more than 1 LNB.
- "symbolrate", this is the symbol rate per seconds, for example 22.000.000.
- "frequency", at which receiving datas, like 12.640.000.
- "ttk", the 22 kHz signal, almost always to 1.
- "diseqc", used to control the diseq, (it means only when you have a diseq controlling 2 or more converters) 0 if none.
- "AFC", usually set to 1.
- "polarization", 1 for horizontal or 0 for vertical one.
- "filter_n <PID> <MAC> <BITFILTER>", where n is intended 0 for Unicast and 1−9 for Multicast; PID is expressed in decimal form, MAC is the MAC address only valid for Multicast filter and BITFILTER is a number which bits represent bytes to mask (for example 10 = 2 (bit 1) + 8 (bit 3), so mask will be 00 00 FF 00 FF 00).

  Example

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

# DVB receiver configuration file, (c) 2000 data planet international

# standard location in /etc

# LNB power on=1/off=0

power 1

# symbol rate [symbol/sec]

symbolrate 22000000

# ASTRA TR 114

frequency 12640000

# 22 kHz signal on=1/off=0

ttk 1

# diseqc on=1/off=0

diseqc 0

# AFC on=1/off=0

AFC 1

# polarisation H=1/V=0

polarisation 1

# settings for MPE filter, PID and MAC filtering, valid MAC bytes

filter_0 512

filter_1 785 00:D0:5C:1E:96:01 48

filter_2 786 00:D0:5C:1E:96:01 48

filter_3 1041 00:D0:5C:1E:96:01 48

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

filter_0 has no MAC and no bitfilter values cause the right MAC address is calculated from IP address (see Appendix A). We will see this setting is OK only for some ISP, for others we'll have to change dvbd.c

## 6.3 Dvbd daemon

Once your /etc/dvbd.conf is ok, you can launch dvbd application, which, if executed without −d option, write to stdout signal quality level:

- Sync MUST be 127 or near it
- Vber MUST be 0

unless you are not well receiving from Sat (check cable and/or dish pointing).

Note:

Maybe you have to change, in dvbd.h this line

#define network_device "eth0"

to

#define network_device "ppp0"

depending on which interface you use to reach Internet, eth0 or ppp0: type "make" to update binary file and restart dvbd.

## 6.4 How to config EON service

Now you have a good signal, you can try to use some sat service.

For EON go at "proxy" setting in Netscape preferences and set under HTTP and FTP:

proxy.xxx.europeonline.net

and, in "port" 8080 and FTP proxy with "port" 8090.

where xxx is the transponder number (103,113,114 or 115) you are using (see Appendix B for more).

Now you should be able to navigate wherever you want.... Good navigation.

To share EON service with many clients you can use  Squid proxy application, enabling cascade to EON proxy.

For a more complex use of EON, like more complex cascade proxy or sharing users, see  EON Linux Masquering FAQ Page

## 6.5 How to use Netsystem service

Netsystem service is a little more complicated than EON under Linux, cause, in addition, you need to setup:

1. VPN connection
2. patch for pppd (needed only if using pppd <= 2.4.0)
3. routing settings with simple scripts
4. patch for dvbd.c
5. Test it
6. Get performance better
7. Sharing Netsystem with many clients

### VPN connection

First you need to download  VPN PPTP client application.

After untared, compiled and installed it, you should add an entry to your /etc/ppp/pap−secrets and /etc/ppp/chap−secrets files, like that:

"login" * "password" *

where "login" and "password" are the same according to  Netsystem registration.

### Patch for pppd

As described at  PPTP description, you need to patch your pppd daemon to support connection with Netsystem VPN server (Linux server).

Warning: using pppd version >= 2.4.0 you don't need pppd patch.

So you have to:

1. download a  recent pppd version
2. download and gunzip the correspondent patch for pppd from  Here
3. untar pppd in a directory
4. type "patch −p0 < patch_name"
5. enter pppd directory
6. type "make", "make install"

## Routing settings

Now your pppd will be able to working with:

"pptp vpn.netsystem.com debug user <login>"

where <login> is your login account from Netsystem: you should see, in log file (/var/log/messages) ppp1 connection debug info.

If all is ok you should see ppp1 interface with "ifconfig" command.

If you still have problems on authentication, please add a "noauth" line to your /etc/ppp/options file.

Once ppp1 interface is up, you should do the following:

1. type "ifconfig ppp1" and find the IP address (I will call IP) on the right of "P−t−P:" string.
2. delete it from routing table with "route del IP"
3. add it to ppp0 interface with "route add IP dev ppp0"
4. delete default gateway from ppp0 with "route del default"
5. add default gateway to ppp1 with "route add default dev ppp1"

Points 1−3 are requested cause point−to−point interface are managed, under Linux, adding the gateway to the new interface (which is not a good idea in this case): unless it you will have a endless loop, cause your packet will be continuously encapsulated on itself.

Points 4,5 are used to make "all internet requests" to ppp1 interface, so we'll reach the "world" by using VPN connection: this could be not optimal in some condition, for example for DNS queries, which could be sent directly to avoid useless Sat delay time.

Instead of manually setup routing configuration you can try using these little scripts:

"netsystem.on" script

---

route add IP_DNS1 dev ppp0

route add IP_DNS2 dev ppp0

route add −net 212.31.242.0 netmask 255.255.255.0 dev ppp0

pptp vpn.netsystem.com user <login>

/bin/sleep 5

route add default dev ppp1

---

"netsystem.off" script

---

route del IP_DNS1 dev ppp0

route del IP_DNS2 dev ppp0

route del −net 212.31.242.0 netmask 255.255.255.0 dev ppp0

kill −9 `ps x|grep "pppd"|grep "<login>"|grep −v "ps"|tr " " "\n"|head −n 2`

rm −−force /var/lock/LCK..tty*

rm −−force /var/run/pptp/*

rm −−force /var/run/ppp1.pid

killall −9 pptp

---

IP_DNS1 and IP_DNS2 are ip addresses of your dns servers (primary and secondary).

<login> is the login name of your Netsystem account.

I tried it out under kernel 2.4.6 RedHat 7.1 and it works very well (without any problems about ppp1 endless loop or similar).

This line:

kill −9 `ps x|grep "pppd"|grep "<login>"|grep −v "ps"|tr " " "\n"|head −n 2`

is used to find PID of pppd process talking with VPN server (ppp1 interface): notice that you cannot just only type " killall pppd" cause your ppp0 interface also would go down.

## Patch for dvbd.c

After solved problems about PPTP you have to change some line in dvbd.c, near the end of it:

```
if (strcmp (v, "filter_0") == 0) { if (s != NULL) { unsigned char ip[4];
dvbcfg[0].status = ON ;
dvbcfg[0].filter.data[0] = 0x3eff ;
dvbcfg[0].filter.pid = (__u16) atoi (s) ;
dvbcfg[0].filter.mode = 0x0c ;
if (ipget (ip, network_device)) { fprintf(stderr,"Can't get local ip address. Stop.\n") ; return -1
syslog (LOG_NOTICE, "Local ip is %u:%u:%u:%u\n", ip[0], ip[1], ip[2], ip[3]);
dvbcfg[0].filter.data[1] = (ip[3] << 8) | 0x00ff ;
dvbcfg[0].filter.data[2] = (ip[2] << 8) | 0x00ff ;
dvbcfg[0].filter.data[6] = (ip[1] << 8) | 0x00ff ;
dvbcfg[0].filter.data[7] = (ip[0] << 8) | 0x00ff ;
dvbcfg[0].filter.data[8] = (0x02 << 8) | 0x00ff ;
dvbcfg[0].filter.data[9] = (0x00 << 8) | 0x00ff ;
setmac (ip) ; }
else { dvbcfg[1].status = OFF ; } }
```

Now following lines:

dvbcfg[0].filter.data[1] = (ip[3] << 8) | 0x00ff ;

dvbcfg[0].filter.data[2] = (ip[2] << 8) | 0x00ff ;

dvbcfg[0].filter.data[6] = (ip[1] << 8) | 0x00ff ;

dvbcfg[0].filter.data[7] = (ip[0] << 8) | 0x00ff ;

dvbcfg[0].filter.data[8] = (0x02 << 8) | 0x00ff ;

dvbcfg[0].filter.data[9] = (0x00 << 8) | 0x00ff ;

will be changed to

dvbcfg[0].filter.data[1] = (MAC[5] << 8) | 0x00ff ;

dvbcfg[0].filter.data[2] = (MAC[4] << 8) | 0x00ff;

dvbcfg[0].filter.data[6] = (MAC[3] << 8) | 0x00ff ;

dvbcfg[0].filter.data[7] = (MAC[2] << 8) | 0x00ff ;

dvbcfg[0].filter.data[8] = (MAC[1] << 8) | 0x00ff ;

dvbcfg[0].filter.data[9] = (MAC[0] << 8) | 0x00ff ;

Where MAC[0]:MAC[1]:MAC[2]:MAC[3]:MAC[4]:MAC[5] is our MAC address (according to Netsystem registration).

For example, using the address 00:d0:d0:d0:d0:d0 we'll have:

dvbcfg[0].filter.data[1] = (0xd0 << 8) | 0x00ff ;

dvbcfg[0].filter.data[2] = (0xd0 << 8) | 0x00ff;

dvbcfg[0].filter.data[6] = (0xd0 << 8) | 0x00ff ;

dvbcfg[0].filter.data[7] = (0xd0 << 8) | 0x00ff ;

dvbcfg[0].filter.data[8] = (0xd0 << 8) | 0x00ff ;

dvbcfg[0].filter.data[9] = (0x00 << 8) | 0x00ff ;

notice hexadecimal notation 0x??

After, you have to type "make" and use the new dvbd created.

Note: to successfully patch the dvbd.c you need to use dvb driver version >= 0.8.2, cause older versions have some instability problems.

## Test it

Finally, we can test Netsystem under Linux. We can make a "ping www.somehostpingable.com" and check the response time: it should be between 400 and 2000 ms.

If you still have problems, you should control if all is OK with VPN interface:

1. open your preferred network sniffer (for example Ethereal) and starts to analyze "ppp0" interface (ppp0, not ppp1!!)
2. make a ping

If VPN is ok you should see 2 (or maybe 1) GRE−Encapsulated packets each second, endless. If you cannot view anything your VPN is not correctly working: stop it and restart it.

## Get performance better

Once you setup all things you NEED to use (particularly with Netsystem service) some "download accelerator" to get performance better: please see Appendix A for more.

## Sharing Netsystem with many clients

To do this you can enable "IP Masquering", allowing your client to use VPN like a normal Internet interface; main problem is that our satellite connection is very good for download while it has bad performance for just browsing web pages (or other service more interactive than downloading).

You can think to use Squid proxy or Socks proxy, but you don't solve your problems, cause even now ALL your request would be forwarded to same interface, VPN.

The solution is to use 2 routing tables, one using direct line interface and the other using VPN one. So you can do like this:

1. be sure of have installed "iproute2" commands (for example type "ip" on shell and control if it tells you something), for more see Linux 2.4 Advanced Routing HOWTO.
2. be sure of have started Netsystem service and annotate ppp1 interface IP address, we'll call it LOCALIP.
3. type: "echo "210 sat" >> /etc/iproute2/rt_tables", to more comfortable call "sat" rule 210
4. type: "ip rule add from LOCALIP table sat", to create table "sat" relative to all requests coming from LOCALIP IP address.
5. type: "ip route add default dev ppp1 table sat", to send all "sat" requests (see above) to ppp1 interface.
6. if you use Socks proxy be sure to set, in sockd.conf file, "external" to LOCALIP.
7. if you use Squid proxy be sure to set, in squid.conf file, "tcp_outgoing_address" to LOCALIP.

Once done all that, you will notice to have 2 kind of working: without any proxy your clients will ask to direct line, while by using proxy (squid or sockd) the request will be forwarded to VPN interface and, definitely, toward satellite.

Notice that maybe you wish to use sockd instead of squid, cause satellite requests are typically used for download (while squid is typically used for browsing...).

What happens with iproute2 commands is that, when you ask for an address to sockd or squid, relative proxy (using IPLOCAL IP address, bound at run−time by proxy network daemon) request enters the TCP/IP stack where kernel will forward it (thanks to point 4 above) to sat table and, definitely (by using point 5) to ppp1 interface. All other rules will be forwarded to classical default route (I mean across ppp0 interface or whatever other interface for direct Internet).

## 6.6 How to use Sat Node service

You have to follow all instructions as for Netsystem.

Before enabling VPN connection, you need to type:

- "route del default", we delete default route
- "route add 212.56.224.36 dev ppp0", we tell it that it has to reach the vpn server just through ppp0
- "pptp 212.56.224.36 user user−name", we create VPN
- "route add default dev ppp1", we tell it that it must use this route to reach every place.

What really changes from Netsystem is that, we don't force VPN gateway (212.56.224.34, IP on the right of "P−t−P" in ppp1 interface) on ppp0 interface, but we force another IP (212.56.224.36). All other things should not change.

Thanks to Ricardo Santiago Mozos and Norberto Garcia Prieto.

## 6.7 OpenSky service under Linux

1. You need to register at  Opensky Registration site. You'll receive a confirming email with login and password in few days.
2. After you have to go to  https://sdr.eutelsat.net and enter login and password data to connect.
3. Set the proxy: "proxy.eutelsat.net:8080".

It is strongly suggested to use downloader applications (see Appendix A for more) to get performance better.

OpenSky is the latest satellite service and it offers 300 MB at month (for free).

Configuration is pretty like EON service, you have to use 0.8.2 siemens drivers you download from LinuxTV, then you NEED to patch dvbd applications.

To apply the patch and to test OpenSky you can find useful infos at:

- http://members.tripod.com/andrei_b/linux/dvbd/dvb−app−dvbd.htm
- http://pguiral.free.fr/DVBS/opensky.html.

---

# 7. Configuration under Windows

## 7.1 EON

Hauppage WinTV has DVB−DATA application that allows to specify data−link settings.

## 7.2 Netsystem

First you need to install VPN capabilities.

In addition you need to download Netsystem software (always foreground spot banner) and launch it: after you should not be able to use Netsystem service: you can download it from here

## 7.3 OpenSky

See Linux.

---

# 8. Appendix A − Notes

## 8.1 Translation Dynamic IP −> MAC Address

The translation used by some ISPs to calculate MAC address (which need to have your DVB card to receive their packets) is:

00 : 01 : IP[0] : IP[1] : IP[2] : IP[3]

where

IP[0].IP[1].IP[2].IP[3] is your dynamic IP address.

This feature is used, for example, by EON.

## 8.2 TCP Window

Satellite connections are an interesting example of very high RTT (round trip time, access time): another example is the Mars − Earth communication or also the Moon − Earth one.

These connections have a very bad feature: very low interactively.

Typical network (or digital, generally) connections use the so called transmission "window", which represents the data buffer can be sent before waiting for an answer. In TCP/IP protocol stack this is the TCP Window.

```
              ----------------------
| - - - - - - - > can continue |-|-|-|---->
|             ----------------------
|            Buffer sendable before confirm
|
| - - - - - - - <----------------------
               Confirmation Answering
```

Now, if our communication has an high access time and if we had a little TCP Window we would lose very much time only waiting for the data confirmation (ACK), so the real bandwidth would decrease (for example if you have a 16KB TCP window, typical of Windows systems and a RTT of 400 ms, you cannot overcome

16KB/0.4 = 40 KB/s)

Solution is to use a very high TCP Window (such as 256 KB or some MB).

Unfortunately, under many systems, is not so simple to have a great TCP Window, so, in latest years, it starts to appear new applications ("download accelerators" described in next section) that split in many pieces a file and download all them in the main time: this is just equivalent to download only one file with a single piece size, avoiding the RTT problem.

## 8.3 Download accelerator

We report here some useful link to so called "download accelerator" which is an application that does 2 things:

1. resuming session management, it allows stopping downloading and restarting later (this can be done thanks to "resuming feature" introduced in Http and Ftp server, which allows you to specify at what byte number start to download).
2. splitting, divide the file in many pieces and let them start in a multithreaded fashion.

As we saw in the previous section, a download accelerator allows us to increase satellite bandwidth.

### Under Linux

- Aria

### Under Windows

- FlashGet
- GetRight
- Mass Downloader

---

# 9. Appendix B − Well Known Sat ISP setting value

## 9.1 EuropeOnLine

EON sends data from Astra satellite (19.2 SE).

MAC address is calculated from IP address (see Appendix A).

It uses "Proxy Authentication".

Follows the 4 transponder setting:

### Transponder 113

- Frequency: 12633.250 MHz,
- SRate : 22 MS/s

- Polarization: Horizontal
- Unicast PID: 512 (decimal)

### Transponder 114

- Frequency: 12640 MHz,
- SRate : 22 MS/s
- Polarization: Vertical
- Unicast PID: 512 (decimal)

### Transponder 115

- Frequency: 12662.750 MHz,
- SRate : 22 MS/s
- Polarization: Horizontal
- Unicast PID: 512 (decimal)

### Transponder 103

- Frequency: 12461 MHz,
- SRate : 27.5 MS/s
- Polarization: Horizontal
- Unicast PID: 512 (decimal)

## 9.2 Netsystem

Netsystem uses Astra satellite (19.2 SE) to send data .

MAC address used is your real MAC address DVB card.

It uses VPN connection.

Follows data setting:

### Transponder 119

- Frequency: 12721 MHz
- SRate: 22MS/s
- Polarization: Horizontal
- Unicast PID: 451 (decimal)

## 9.3 Sat Node

Sat Node uses Astra satellite (19.2 SE).

- Frequency: 12603.750 MHz
- SRate: 22MS/s
- Polarization: Horizontal
- Unicast PID: 302

## 9.4 Open Sky

Open sky uses Eutelsat satellite (7 SE).

- Frequency: 11262 MHz
- SRate: 27.5MS/s
- Polarization: Horizontal
- Unicast PID: 3003

# 10. Links

## 10.1 Open

http://www.linuxtv.org

http://www.fatamorgana.com/bertolinux http://www.hack−it.net/How−To/Sat−HOWTO.html

## 10.2 Commercial

EuropeOnLine − EON

Netsystem

Starspeeder

Sat Node

Eliosat

Falcon Stream

SkyDSL

OpenSky

DirecPC

http://www.xanticbroadband.com/