

SelfLinux-0.10.0



TDSLviaSAT-HowTo

Autor: Wolfgang Wershofen (*itconsult@wershofen.de*)

Formatierung: Torsten Hemm (*T.Hemm@gmx.de*)

Lizenz: GFDL

Inhaltsverzeichnis

1 Einleitung

- 1.1 Sinn und Zweck dieses Dokuments
- 1.2 Copyright
- 1.3 Übersetzungen
- 1.4 Danksagung

2 Wissenswertes vor dem Start

- 2.1 T-DSL via Satellit?
- 2.2 Link-Sharing
- 2.3 vorausgesetztes Grundwissen
- 2.4 Empfehlenswerte Lektüre
- 2.5 1.1. Informationsquellen im Netz

3 Systemvoraussetzungen

- 3.1 Hardware
- 3.2 Treiber und Kernel
- 3.3 konventioneller Internet-Zugang
- 3.4 Software
 - 3.4.1 dvbtune
 - 3.4.2 Tellique Client
 - 3.4.3 pptp-Client

4 Installation

- 4.1 Vorbemerkung: root oder nicht root - das ist hier die Frage
- 4.2 Treiber
- 4.3 dvbtune-Installation
- 4.4 Tellique-Proxy

5 Das erste Mal ...

- 5.1 Registrierung bei der Telekom
- 5.2 Laden der Treiber
- 5.3 Einstellen der richtigen Frequenz
 - 5.3.1 Tabelle der Parameter
- 5.4 Starten des Netzwerk-Interfaces
- 5.5 Testen der Verbindung
- 5.6 Starten des Proxy
- 5.7 Einstellung Proxy am Browser
 - 5.7.1 Lynx
 - 5.7.2 Konqueror
 - 5.7.3 Netscape/Mozilla
- 5.8 Erster Login auf dem Proxy-Server
- 5.9 Herzlichen Glückwunsch!

6 Die Alternative:

- 6.1 Unterschiede zwischen VPN und Proxy
- 6.2 Installation pptp-Client
- 6.3 Benutzerkennung in /etc/ppp/pap-secrets und /etc/ppp/chap-secrets

- 6.4 Erster Verbindungstest
- 6.5 Konfiguration DVB-Device
- 6.6 Verändern des Routings
- 6.7 Zweiter Test


7 Tips, Tricks, Bugs und was sonst noch so dazu gehört

- 7.1 Automatisierung
 - 7.1.1 Initialisierung beim Wechsel des Runlevels
 - 7.1.2 Initialisierung beim Aufbau der Internet-Verbindung
 - 7.1.3 Anpassungen /etc/ip-up Script für VPN-Zugang
- 7.2 Zugang mit ISDN - Dial on Demand
- 7.3 Einsatz im Netzwerk
- 7.4 Bugs und Einschränkungen
- 7.5 Ausblick

1 Einleitung

1.1 Sinn und Zweck dieses Dokuments

Dies ist mein erstes selbstverfasstes HowTo. Ich bitte darum, mich über evtl. vorkommende handwerkliche Fehler oder Unstimmigkeiten in Kenntnis zu setzen. Als Vorlage für Layout und Aufbau habe ich das SAT-HOWTO von *Roberto Arcomano* und *Florindo Santoro* genutzt. Der Grund, warum ich dieses HowTo verfasst habe, ist zum einen, für mich selbst eine Dokumentation über die Konfiguration und Funktionsweise des Satelliten-DSLs der *Deutschen Telekom* unter Linux zu erstellen, zum anderen aber auch, um anderen Linux-Nutzern dieses Dienstes die nächtelange Kleinstarbeit mit enttäuschten Hoffnungsschimmern, Frustrationen, Selbstzweifeln und dem Bedürfnis "die Kiste einfach aus dem Fenster zu schmeissen", zu ersparen. Außerdem will ich die süße Genugtuung, es letztendlich doch geschafft zu haben, für mich behalten. ;-)

Der Hauptteil der Anleitung befasst sich mit dem Zugang zu T-DSL via Satellit über den Telligence-Proxy und Multicast-Routing. Auf Seite 6 des Dokuments finden Sie aber auch die Beschreibung, wie eine Unicast-Verbindung über ein VPN (Virtual Private Network) aufgebaut werden kann. Dieses Dokument ist das Produkt meiner privaten Initiative und in keinsten Weise von der *Deutschen Telekom* in Auftrag gegeben oder gar bezahlt worden. Ich freue mich über jede Art des Feedbacks, sei es Lob oder Tadel. Letzteres hoffentlich dann in konstruktiver Form, damit dieses Dokument tatsächlich eine brauchbare Anleitung für den Neuling in Sachen T-DSL via Satellit unter Linux wird. Zu erreichen bin ich unter der EMail-Adresse: 

1.2 Copyright

Copyright (C) 2002 *Wolfgang Wershofen*. Dieses Dokument ist frei. Sie können es unter den Bedingungen der *GNU General Public License*, Version 2, wie von der *Free Software Foundation* herausgegeben, weitergeben und/oder modifizieren. Die Veröffentlichung dieses Dokumentes erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE JEDE GEWÄHRLEISTUNG - sogar ohne die implizite Gewährleistung der MARKTREIFE oder der EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. Details finden Sie in der *GNU General Public Licence*.

1.3 Übersetzungen

Mir sind bislang noch keine Übersetzungen dieses Dokumentes bekannt. Wenn es gewünscht wird, wäre ich bereit, eine englischsprachige Übersetzung zu erstellen. Sofern Sie Interesse haben sollten, selbst eine Übersetzung in einer anderen Sprache anzufertigen, sollten Sie an drei Dinge denken:

1. Prüfen Sie, ob es nicht schon eine Übersetzung in Ihrer Sprache gibt. Wenden Sie sich dazu an ihr lokales Linux Documentation Project.
2. Pflegen Sie bitte diesen 1. Abschnitt des Dokumentes entsprechend
3. Informieren Sie mich über die Übersetzung, damit ich Sie in einer Liste unter diesem Punkt aufnehmen kann.

Vielen Dank für Ihre Übersetzung.

1.4 Danksagung

Es gibt einige Leute, ohne die dieses Dokument niemals entstanden wäre oder aber wesentliche Verbesserungen und Hinweise fehlen würden. Ich möchte mich an dieser Stelle bei all jenen für ihre Unterstützung bedanken.

Ralph und Marcus Metzler für die hervorragende Programmierung der DVB- Treiber und offene Ohren bei Problemen Fa. Convergence, die sich auf die Fahnen geschrieben hat, die DVB-Treiber weiterhin als Open-Source unter der GNU-GPL zu veröffentlichen und weiterzuentwickeln.

Roberto Arcomano und Florindo Santoro für das gelungene SAT-HowTo

Dave Chapman (bitte EMails nur in Englisch) und *Hilmar Linder* für die Entwicklung von dvbtune und Hilfe bei Problemen


Andrei Boros für manche beantwortete Frage und seine sehr informative Homepage

Dirk Wellmann für die Assistenz in der Test- und Konsolidierungsphase sowie einige Textpassagen

Carsten Koch für's Korrekturlesen

Markus Dahlweid für die Anpassung der Init-Skripte für SuSE 8.0 *Thierry Coutelier* von SES ASTRA für den entscheidenden Hinweis bei der Konfiguration des VPN-Zugangs.

last not least:

Roland Klabunde für's Aufwecken, die richtigen Fragen, noch bessere Antworten und sein Know-How sowie allen anderen Betreibern von  <http://www.ipviasky.net>. Danke für Eure Initiative.

2 Wissenswertes vor dem Start

2.1 T-DSL via Satellit?

Seit Mai 2002 bietet die *Deutsche Telekom* einen Satelliten-basierten Internet-Zugang mit T-DSL-Geschwindigkeit von bis zu 768kb/s im Downstream an. Dieser wurde zuvor in einer 6 monatigen Pilotphase mit 500 ausgewählten Teilnehmern getestet. Der Zugang erfolgt über den Satelliten Astra 19.2° und ist hauptsächlich für Telekom-Kunden gedacht, deren Haushalte nicht mit dem "normalen" DSL via Kabel versorgt werden können. Ein großer Vorteil bei der Wahl des Satelliten ist übrigens, dass dies der gleiche Satellit ist, über den auch die ASTRA- Fernsehprogramme empfangen werden können. Es muß also (im Gegensatz zu manch anderen SAT-IP-Anbietern) keine zusätzliche SAT-Antenne installiert werden, um Fernsehen und Internet über den Satelliten zu empfangen. Bei T-DSL via Satellit erfolgt die Datenübertragung in's Internet (upstream) weiterhin wie üblich über Modem oder ISDN, der Rückweg (downstream) erfolgt dann über Satellit. Insofern fallen also für die Einwahl zusätzliche Kosten für die Telefonverbindung an. Die Telekom bietet derzeit zwei verschiedene Tarifvarianten an. Die erste (und billigere) Variante bietet ein Volumen von 500MB pro Monat zum Festpreis an, jedes weitere MB darüber hinaus wird gesondert abgerechnet. Variante 2 bietet ebenfalls 500 MB Traffic an, allerdings werden die darüber hinausgehenden Transfers nicht gesondert berechnet, sondern lediglich mit einer niedrigeren Priorität - sprich u.U. langsamer - transferiert. Die Telekom nennt dies "intelligentes Bandbreitenmanagement". Näheres zu den angebotenen Tarif-Varianten finden Sie auf den Webseiten der Deutschen Telekom.

2.2 Link-Sharing

Ein interessanter Aspekt von T-DSL via Satellit ist das sogenannte Link-Sharing. Bei vielen anderen Satelliten-Diensten für den Internet-Zugang, wird die terrestrische Telefonverbindung tatsächlich nur für den "Hinweg" der Datenpakete in's Netz genutzt, während die Satelliten-Verbindung allein den Rücktransport der angeforderten Daten übernimmt. Dies hat zwei entscheidende Begleiterscheinungen:

1. Solange nur kleine Datenmengen, wie z.B. beim normalen Surfen im Internet, übertragen werden, ist die subjektive Übertragungsgeschwindigkeit langsamer, da alle Datenpakete den weiten Weg über den Satelliten gehen müssen und somit mit einer Verzögerung von ca. 0,5 Sekunden am Zielrechner ankommen.
2. Da tatsächlich alle Daten über den Satelliten gehen, ist die Bandbreite des Transponders schnell ausgeschöpft und die Übertragungsgeschwindigkeit geht in den Keller. Zu Stoßzeiten kann es daher vorkommen, dass die Übertragungsrate über den Satelliten deutlich unter die der herkömmlichen Telefonverbindung sinkt. Diesen beiden Effekten versucht T-DSL via Satellit mit dem sogenannten Link-Sharing entgegen zu wirken. Beim Link-Sharing wird neben dem Satelliten auch die terrestrische Telefonverbindung für den Rücktransport der Daten genutzt. Erst wenn die Bandbreite dieser Verbindung nicht mehr ausreicht, um die Daten ohne Verzug zum anfordernden Rechner zurückzubringen, wird der Weg über den Satelliten eingeschaltet, um den Datentransfer zu beschleunigen. Dies führt auf der einen Seite zu einer Entlastung des Satelliten, weil ein großer Teil der Datenmenge gar nicht über den Satelliten läuft. Für den Nutzer bietet dies auf der anderen Seite den Vorteil, dass die Reaktionszeit, bis die angeforderten Daten bei ihm ankommen, im gewohnten Rahmen bleiben und die Übertragungsrate nicht unter die Performance der Telefonverbindung sinken kann. Das Link-Sharing wird seit einer Änderung der VPN-Server Anfang August nun auch für den VPN-Zugang genutzt und macht somit diesen Zugang zu einer tatsächlichen Alternative für den Proxy.

2.3 vorausgesetztes Grundwissen

Dieses Dokument konzentriert sich auf die Belange des T-DSL via Satellit-Angebotes der Deutschen Telekom und stellt keine generelle Dokumentation für die Konfiguration des Satelliten-Zugangs an sich dar. Es wird

davon ausgegangen, dass die generelle Verbindung zum Satelliten funktioniert und ein gewisses Grundwissen im Betrieb des Systems unter Linux vorhanden ist. Sollten Sie Zweifel haben, über das geforderte Grundwissen zu verfügen, empfehle ich Ihnen das eingehende Studium der im Folgenden genannten Quellen.

2.4 Empfehlenswerte Lektüre

SAT-HowTo von Roberto Arcomano und Florindo Santoro

Dieses Dokument ist ein "Muß" für den Network-Zugang über Satellit. Es werden einige grundlegende Dinge erklärt und auch alternative Anbieter sowie deren Konfiguration genannt. Zur Zeit (Juni 2002) beschäftigt sich dieses Dokument leider nur mit der Funktionsweise der Version 0.8.x der DVB-Treiber. Eine Aktualisierung für die Treiber der neuesten Generation ist aber in nächster Zeit vorgesehen.


FAQ von IP via Sky


Hier werden eine Menge Informationen speziell über den T-DSL via Satellit-Dienst gegeben - zwar etwas Windows-lastig aber die grundlegende Funktionsweise und Einstellungen sind für Linux unverändert übertragbar.


2.5 1.1. Informationsquellen im Netz

Weitere Informationen rund um den Satelliten-Zugang über Linux finden sich im Internet. Hier eine kleine Auswahl:

 <http://www.linuxtv.org> die offizielle Seite der Treiberentwickler von Convergence.

 <http://www.linuxtv.org/maillinglists/linux-dvb> Interessante (englischsprachige) Mailingliste rund um DVB unter Linux mit durchsuchbarem Archiv. Für Probleme mit den Treibern die erste Anlaufadresse. Über diesen Link kann man die Mailingliste abonnieren www.linuxdvb.tv Hier findet man täglich aktuell Snapshots mit den aktuellen CVS-Versionen der Treiber

 <http://www.linuxstb.org> Seite von Dave Chapman, dem Entwickler von dvbtune. Dort gibt es noch eine Menge mehr nützlicher, kleiner Tools rund um die DVB-Karte

 <http://www.ipviasky.net> Portal von Mitarbeitern der Deutschen Telekom, die den Satelliten-Dienst mit gestaltet haben. Hier erhält man alle Informationen rund um den Zugang aus erster Hand von den Leuten, die es wissen müssen. Mit FAQ und Newsforum sowie Download-Area.

Homepage von Andrei Boros Sat-Netzwerk-Zugang mit 0.8.x-Treibern. Sehr informativ.

Homepage von Hubertus Sandmann


Homepage mit vielen Informationen rund um Linux, DVB allerdings mit Schwerpunkt digital-TV und Harddisk-Recording.

Diese Liste erhebt keinen Anspruch auf Vollständigkeit. Wenn jemand noch weitere zum Thema passende und interessante Links kennt, bitte zumailen.

3 Systemvoraussetzungen

3.1 Hardware

Für die Teilnahme an T-DSL via Satellit unter Linux werden folgende Hardwarekomponenten benötigt: PC mit Linux-Betriebssystem und Modem oder ISDN-Karte bzw. -Terminaladapter Satelliten-Antenne mit digitaltauglichem LNB DVB-PCI-Steckkarte


Wie schon eingangs erwähnt, soll dieses HowTo nicht die komplette Konfiguration und Installation des Systems abdecken, sondern sich auf die Spezialitäten des T-DSL-Angebotes konzentrieren. Es sollte jedoch klar sein, dass die hier beschriebene Vorgehensweise nur funktioniert, wenn die Hardware entsprechend konfiguriert und installiert ist. Überlassen Sie im Zweifelsfall einem Fachmann das Aufstellen und Ausrichten der Antennenanlage sowie die Verkabelung der einzelnen Komponenten. Die Satelliten-Antenne muss auf den ASTRA-Satelliten, Orbitalposition 19.2 Grad ausgerichtet sein. Die Schnittstelle zwischen Antenne und PC stellt die DVB-Karte dar. Diese PCI-Steckkarte ist zwar primär für den Empfang von digitalen Radio- und Fernsehprogrammen gedacht, kann aber auch für den Empfang von IP-Paketen über den Satelliten genutzt werden. Achten Sie beim Kauf darauf, eine Karte mit einem von den Linux-Treibern unterstützten Chipsatz zu kaufen. Informationen, welche Karten derzeit unter Linux laufen, erhalten Sie bei  <http://LinuxTV.org>. Derzeit sind zwei Varianten der DVB-Karten gebräuchlich. Zum einen die vollwertigen DVB-s Karten mit integriertem MPEG2-Decoder (z.B. Hauppauge WinTV DVBs) und zum anderen die "budget"-Karten ohne MPEG-Decoder (Hauppauge WinTV nova und Technotrend budget PCI). Grundsätzlich sind beide Typen für das IP-Networking via Satellit geeignet, allerdings gibt es derzeit bei den vollwertigen Karten mit MPEG2-Decoder noch einen Bug der beim Proxy-Zugang mit Multicasting zu regelmäßigen Abstürzen des LINUX-Systems führt. Stabil läuft TDSLvSAT über Multicasting derzeit nur mit den "budget"-Karten. Als stabile Alternative für die Karten mit MPEG-Decoder bietet sich der VPN-Zugang über Unicast an. Als optimale Lösung würde ich die Kombination von zwei DVB-Karten erachten - eine "vollwertige" DVB-s mit MPEG-Decoder für digitales Fernsehen und DVD in hervorragender Qualität und eine "Budget"-Karte für den Internet-Zugang. Mit einer solchen Lösung können Sie dann auch parallel surfen und fernsehen.

Ich selbst verwende für den Satelliten-Empfang folgende Hardware: 85 cm Satellitenantenne mit digitalem Doppel-LNB Technotrend budget PCI DVB-Karte PC mit Elitegroup K7S5A-Motherboard, AMD Thunderbird 1.4 GHz Prozessor, 512 MB RAM, ATI Radeon ViVo 64MB DDR Grafikkarte

Sowohl der Netzwerkzugriff als auch Digital-TV und Radio funktionieren damit problemlos.

3.2 Treiber und Kernel

Im Februar 2001 haben die Entwickler der Linux-DVB-Treiber einen harten Schnitt in der Architektur der Treiber vollzogen, um sich an die geänderten Gegebenheiten der neuen Linux-Kernel-Versionen 2.4.x anzupassen. Dies hat zur Folge, dass die bis dahin entwickelten Treiber-Versionen bis einschließlich 0.8.2 nur auf den Kernen 2.2.x laufen und für Kernel 2.4.x und folgende die Treiber 0.9.x verwendet werden müssen. Da die 0.8.x-Treiber nicht mehr weiterentwickelt werden und ich selbst einen 2.4.16er Kernel einsetze, wird die Konfiguration des SAT-IP-Zugangs nur für die 0.9er-Generation der Treiber beschrieben. Wer in seinem System allerdings immer noch einen 2.2.x-Kernel laufen hat und sich vor einem Kernel-Update scheut, sollte sich die Seiten von Andrei Boros einmal genauer anschauen. Dort wird der Zugang mit der älteren Kernel- und Treiberversionen detailliert beschrieben. Die Treiber für die DVB-Karte werden als Source-Tarball ausgeliefert. Sie benötigen daher auf jeden Fall die Sourcen des aktuell verwendeten Kernels um die Treiber kompilieren zu können. Manche Distributionen installieren diese nicht automatisch mit, weshalb Sie vorher prüfen sollten, ob Sie diese evtl. nachinstallieren müssen. Normalerweise befinden sich die Kernel-Sourcen im Verzeichnis /usr/src/linux. Im Kernel sollte außerdem I2C und Video4Linux zumindest als Modul enthalten sein. Es ist bei

den Linux-DVB-Treibern immer empfehlenswert, die neueste CVS-Version zu benutzen, da die Entwicklung hier noch sehr dynamisch ist. Die derzeit letzte "offizielle" Version 0.9.4 hat z.B. noch keinen Multicast-Support, der aber für den Zugang zu T-DSL via Satellit in der hier beschriebenen Form unerlässlich ist. Die CVS-Versionen der Treiber erhält man am bequemsten als nächtlichen Snapshot bei  <http://www.linuxdvb.tv>. Der verwendete Treiber-Snapshot sollte nicht älter als der 26. Februar 2002 sein, weil an diesem Datum der Multicast-Support realisiert wurde. Nutzer einer Budget-Karte sollten sogar nicht weiter als bis zum 18. März 2002 zurückgehen, weil an diesem Tag ein schwerer Bug aus den Treiber entfernt wurde, der zum Systemstillstand bei der Konfiguration des Netzwerk-Devices führte. Am 7. April 2002 wurden auch noch einige kleinere Verbesserungen am Netzwerk-Teil der Treiber eingebaut, weshalb momentan ein Treiber nach diesem Datum empfehlenswert ist. Ich selbst verwende die CVS-Version der Treiber vom 21. Mai 2002 mit einem 2.4.16er-Kernel aus der SuSE-7.3 Distribution. Bei Treiber späteren Datums (Ende Juli/Anfang August) kann es Probleme mit dem Tuning über dvbtune geben. Sollte dies bei Ihnen der Fall sein, versuchen Sie zunächst die Treiberversion vom 21. Mai zu verwenden, die garantiert mit dvbtune funktionieren.


3.3 konventioneller Internet-Zugang

Da es sich bei dem Satelliten-Zugang der Deutschen Telekom lediglich um einen "One-Way-Service" handelt, über den Daten lediglich empfangen aber nicht gesendet werden können, ist ein funktionierender Internet-Zugang über Modem, ISDN o.ä. unbedingt erforderlich. Sollten Sie einen solchen Zugang noch nicht haben, richten Sie diesen bitte zuerst ein (obwohl ich mich in dem Falle eigentlich frage, wie Sie überhaupt dieses Dokument lesen können?! ;-)) Hilfestellung bei der Konfiguration finden Sie mit Sicherheit in den Dokumentationen Ihrer Linux-Distribution oder in den HowTo's des Linux Documentation Projects.

3.4 Software

Neben der Hardware und den dazu gehörigen Treibern werden für den Zugang zu T-DSL via Satellit noch die folgenden Programme benötigt:


3.4.1 dvbtune

Die IP-Pakete der T-DSL-Dienstes werden vom Satelliten auf einer bestimmten Frequenz ausgesendet. Auf diese Frequenz muß der LNB über die DVB-Karte eingestellt werden, was von dem kleinen Kommandozeilen-Tool **dvbtune** von *Dave Chapman* erledigt wird. Das Programm ist Open-Source und unterliegt der GNU GPL. Sie können die Quellen von der Seite  <http://www.linuxstb.org> herunterladen. Achten Sie darauf, die Version 0.2 oder 0.3 zu verwenden, weil erst in diesen Versionen der Netzwerk-Support von *Hilmar Linder* enthalten ist. Falls Sie einen LNB-Umschalter (DiSeq) verwenden, sollten Sie auf jeden Fall die Version 0.3 einsetzen, weil diese über eine Steuermöglichkeit des DiSeq verfügt.

3.4.2 Tellique Client

Der Internet-Zugang über T-DSL via Satellit wird über eine Proxy-Lösung realisiert. Der hierzu verwendete Proxy wurde von der Firma *Tellique* entwickelt und kann hier heruntergeladen werden. Das Programm ist Closed-Source und wird als binary zur Verfügung gestellt.

3.4.3 pptp-Client

Sollten Sie beabsichtigen, statt des standardmässigen Proxy-Zugangs die alternative Zugangsmöglichkeit über eine VPN-Verbindung zu nutzen, benötigen Sie den Linux **pptp-Client**, der in der Lage ist, eine VPN-Verbindung zu einem Windows-VPN-Server aufzunehmen. Zu finden ist diese Software unter  <http://pptpclient.sourceforge.net>. Der VPN-Zugang sollte auf jeden Fall für Besitzer einer "full featured" DVB-s

Karte (mit MPEG2-Decoder) erste Wahl sein, da mit diesem die Multicast-Probleme dieser Karten nicht auftreten.

4 Installation

4.1 Vorbemerkung: root oder nicht root - das ist hier die Frage

Grundsätzlich sollte man bei der Arbeit in einem Linux-System so wenig wie möglich mit Superuser-Rechten - also als root - unterwegs sein. Bei einer Aufgabe wie der Software-Installation und u.U. auch beim ersten Testlauf des neuen Systems ist es aber durchaus sinnvoll, wenn man zunächst Probleme durch unzureichende Zugriffsrechte auf Dateien etc. vermeidet und die Aktionen als root durchführt. Am einfachsten ist es dabei natürlich, wenn man sich als Benutzer root auf einer Console einloggt oder sich vor Beginn der Tätigkeiten mittels

```
user@linux /usr/src/ # su
```

und anschließender Eingabe des root-Passwortes zum Superuser macht. In diesen Fällen ist man allerdings permanent root und sollte daher besondere Vorsicht beim Umgang mit dem System walten lassen. Besser - wenn auch etwas aufwendiger - scheint mir die von mir auch im Folgenden dargestellte Lösung zu sein, nur jeweils die auszuführenden Befehle mit root-Rechten zu versehen, an sich aber als ganz normaler User im System angemeldet zu sein. Wer dennoch eher die bequeme Lösung favorisiert, muß lediglich das vorangestellte `su -c` und die Hochkommata um den eigentlichen Befehl herum weglassen.

4.2 Treiber

Nachdem Sie sich eine passende CVS-Version der DVB-Treiber aus dem Internet heruntergeladen haben (siehe auch Punkt 3.2), entscheiden Sie sich zunächst für ein passendes Verzeichnis, in dem Sie den Tarball auspacken wollen. Da es sich um einen Source-Tarball handelt, wäre z.B. /usr/src ein geeigneter Platz. Bei der folgenden Installationsbeschreibung gehe ich davon aus, dass der Tarball im Home-Verzeichnis liegt und die Installation in /usr/src erfolgt. Mit

```
user@linux ~/ # cd /usr/src
```

wechseln Sie also zunächst in das gewünschte Verzeichnis. Anschließend entpacken Sie den Tarball mit

```
user@linux ~/ # su -c "bunzip2 -c ~/dvb-20020331.tar.bz2 | tar -xvf -"
```

Es wird ein neues Verzeichnis DVB angelegt, in dem die Sourcen und einige Anweisungen enthalten sind. Eine detaillierte Installationsanweisung finden Sie in der Datei /usr/src/DVB/INSTALL. Lesen Sie sich diese Anleitung durch und folgen Sie den dortigen Anweisungen. Sofern Sie allerdings eine "Standard"-Installation haben, sollte es reichen folgende Schritte auszuführen:

Wechseln Sie in das Verzeichnis DVB/driver

```
user@linux /usr/src// # cd /DVB/driver
```

Kompilieren Sie die Treiber mit

```
user@linux /usr/src/DVB/driver/ # su -c make
```

Es sollten keine Fehlermeldungen auftreten. Auftauchende Warnungen können ignoriert werden.

Anschließend können Sie mit Root-Berechtigung die Treiber laden

```
user@linux /usr/src/DVB/driver/ # su -c "make insmod"
```

Auch hierbei sollten keinerlei Fehlermeldungen auftreten. Außerdem sollten in /var/log/messages einige Meldungen zum Ladeprozess zu finden sein. Hier ein Beispiel, wie es bei mir mit der Technotrend budget PCI aussieht:

```
                                /var/log/messages

Mar 22 15:07:47 athlon kernel: i2c-core.o: i2c core module
Mar 22 15:07:47 athlon kernel: Linux video capture interface: v1.00
Mar 22 15:07:48 athlon kernel: i2c-core.o: driver VES1893 DVB
demodulator registered.
Mar 22 15:07:48 athlon kernel: i2c-core.o: driver VES1820 DVB
demodulator registered.
Mar 22 15:07:48 athlon kernel: i2c-core.o: driver L64781 DVB
demodulator registered.
Mar 22 15:07:48 athlon kernel: initSP8870:
Mar 22 15:07:48 athlon kernel: i2c-core.o: driver SP8870 DVB
demodulator registered.
Mar 22 15:07:48 athlon kernel: i2c-core.o: driver tda8083 DVB
demodulator registered.
Mar 22 15:07:48 athlon kernel: i2c-core.o: driver stv0299 DVB
demodulator registered.
Mar 22 15:07:48 athlon kernel: i2c-core.o: driver i2c TV tuner driver
registered.
Mar 22 15:07:49 athlon kernel: saa7146core.o: saa7146(1): bus:0,
rev:1, mem:0xe481de00.
Mar 22 15:07:49 athlon kernel: SP8870: no SP8870 found ...
Mar 22 15:07:49 athlon kernel: i2c-core.o: client [stv0299] registered
to adapter [saa7146(1)](pos. 0).
Mar 22 15:07:49 athlon kernel: tuner: chip found @ 0x61
Mar 22 15:07:49 athlon kernel: i2c-core.o: client [i2c tv tuner chip]
registered to adapter [saa7146(1)](pos. 1).
Mar 22 15:07:49 athlon kernel: i2c-core.o: adapter saa7146(1)
registered as adapter 0.
Mar 22 15:07:50 athlon kernel: dvb: 1 dvb(s) found!
```

Mit

```
user@linux /usr/src/DVB/driver/ # su -c "lsmod"
```

sollten Sie nun einige Module sehen, die mit "dvb" beginnen. Sofern soweit alles funktioniert hat, können Sie mit dem Befehl

```
user@linux /usr/src/DVB/driver/ # su -c "make rmmmod"
```

die Treiber wieder entladen. Die Installation sollte erfolgreich gewesen sein.

4.3 dvbtune-Installation

dvbtune wird ebenfalls als Source-Tarball heruntergeladen. Auch hier empfiehlt sich als Installationsverzeichnis /usr/src. Mit

```
user@linux ~/ # cd /usr/src
```

wechseln Sie zunächst in das gewünschte Verzeichnis.

Anschließend entpacken Sie den Tarball mit

```
user@linux /usr/src / # su -c "tar-xvzf ~/dvbtune-03.tar.bz"
```

Es wird ein neues Verzeichnis dvbtune-0.3 angelegt, in dem die Sourcen enthalten sind. Wechseln Sie in dieses Verzeichnis

```
user@linux /usr/src / # cd dvbtune-0.3
```

und kompilieren Sie das Programm mit

```
user@linux /usr/src/dvbtune-0.3/ # su -c make
```

Anschließend finden Sie das Programm dvbtune in diesem Verzeichnis. Mit

```
user@linux /usr/src/dvbtune-0.3/ # su -c "cp dvbtune /usr/local/bin"
```

kopieren Sie dieses in ein Verzeichnis, dass in Ihrem Pfad liegt. Mehr ist für die Installation des Programms nicht zu tun.

4.4 Tellique-Proxy

Da der Tellique-Proxy nur als binary erhältlich ist, sollte die Installation nicht wie bei den beiden vorherigen Punkten in einem Source-Pfad erfolgen, sondern z.B. unter /opt. Mit

```
user@linux ~/ # somebody@localhost:~ # cd /opt
```

wechseln Sie zunächst in das gewünschte Verzeichnis und legen dort mit

```
user@linux /opt/ # su -c "mkdir tellique"
```

ein neues Verzeichnis, in welches Sie anschließend mit

```
user@linux /opt/ # cd tellique
```


wechseln. Dort entpacken Sie den Tarball mit

```
user@linux /opt/tellique/ # su -c "tar -xvzf ~/proxy222a.tar.gz"
```

Der Tarball enthält drei Dateien. Die eine ist das ausführbare Programm namens **proxy**, die anderen beiden sind ini-Dateien, die zur Parametrisierung des Proxy verwendet werden. Die ini-Dateien - allen voran die `recv.ini` - sind wichtig für den erfolgreichen Datentransfer und müssen entsprechend gepflegt werden. Näheres hierzu im Abschnitt 5.6.

5 Das erste Mal ...

5.1 Registrierung bei der Telekom

Dieser Punkt ist zwar nicht zwingend erforderlich für die Abarbeitung der folgenden Punkte bis einschließlich 5.7., da aber eine Registrierungsbestätigung der *Telekom* benötigt wird, empfiehlt es sich, diesen Punkt zuerst durchzuführen. Unerlässlich für die Nutzung von T-DSL via Satellit ist natürlich, diesen Dienst bei der *Deutschen Telekom* zu bestellen. Sie können dies über die einschlägigen Hotlines, den T-Punkt oder im Web unter  <http://www.telekom.de/t-dslsat> machen. Innerhalb von 5 Werktagen nach Bestellung erhalten Sie einen Brief von der Telekom mit einem vorläufigen Benutzernamen und Passwort (PIN-Brief). Wenn der Brief eintrifft, wird auf obige URL verwiesen, wo Sie dem Link "Registrierung" folgen sollen. Dort werden Sie aufgefordert ihre vorläufige Kennung (User/PWD) einzugeben und haben auch die Möglichkeit, eine eigene User/ Password-Kombination zu wählen. Ein weiteres Pflichtfeld bei der Registrierung - neben Ihrem Usernamen und Passwort - ist die Angabe der MAC-Adresse Ihrer DVB-Karte. Sie finden diese Hardware-Adresse normalerweise auf der DVB-Karte selbst vermerkt. Sollte dies nicht der Fall sein, können Sie die MAC-Adresse ermitteln, sofern Sie auf dem gleichen Rechner auch MS-Windows installiert haben. Mit dem Befehl `winipcfg` können Sie sich die MAC-Adressen aller in Ihrem Rechner enthaltenen Netzwerk-Devices anzeigen lassen. Eine Ermittlungsmöglichkeit unter Linux ist mir derzeit nicht bekannt.

Sollte jemand hierzu einen Tipp haben, wäre ich froh, darüber zu hören. Sollten Sie keine Adresse für Ihre DVB-Karte ermitteln können, dann vergeben Sie selbst eine Adresse im Format `xx:xx:xx:xx:xx:xx`, wobei jedes `xx` eine hexadezimale Zahl zwischen 00 und FF darstellt. Achten Sie darauf, dass sie eine möglichst eindeutige MAC-Adresse verwenden, da ansonsten die Gefahr besteht, dass Sie Datenpakete empfangen, die Sie gar nicht wollen. Falls möglich, verwenden Sie eine MAC-Adresse einer Netzwerkkarte eines anderen PCs, der nach Möglichkeit nicht an das Internet angeschlossen ist. Merken Sie sich die Adresse oder schreiben Sie diese am Besten auf. Sie wird im weiteren Verlauf noch benötigt. Noch ein Hinweis zur MAC-Adresse: Für den Zugang zu T-DSL via Satellit über den Telique-Proxy und Multicasting wird die MAC-Adresse überhaupt nicht gebraucht. Für den alternativen VPN-Zugang ist die Angabe der MAC allerdings unerlässlich, damit sie die angeforderten Datenpakete auch korrekt zurückgeschickt bekommen. Nähere Informationen zum VPN-Zugang finden Sie in Kapitel 6. Seit Anfang August ist es aufgrund von verschiedenen Mißbrauchsversuchen notwendig, die gewünschte Zugangsart (Proxy oder VPN) auf dem Registrierungsbildschirm anzugeben. Sollten sie einen Wechsel Ihrer Zugangsart beabsichtigen, müssen sie dies in Ihren Registrierungsdaten ändern.

5.2 Laden der Treiber

Zunächst müssen die Treiber der DVB-Karte geladen werden. Leider funktioniert dies noch nicht mit **modprobe** automatisch bei Anforderung, sondern muß manuell geschehen. Einen testweisen Ladeversuch habe ich ja schon im Abschnitt 4.1 beschrieben, der jetzige funktioniert genauso. Mit

```
user@linux ~/ # cd /usr/src/DVB/driver wechseln
```

Sie in das DVB-Treiber-Verzeichnis und rufen dort

```
user@linux /usr/src/DVB/driver / # su -c "make insmod" auf. Die
```

Treiber sind anschließend geladen und betriebsbereit.

5.3 Einstellen der richtigen Frequenz

Nachdem nun die Treiber geladen sind, muß nun der LNB auf die richtige Frequenz getuned werden. Allerdings ist die Frequenz nicht der einzige relevante Parameter, sondern auch die Polarisierung, die Signallrate und die sogenannte Multicast-PID (PID = Packet Identification) sind von Bedeutung. Das Tuning übernimmt das Programm **dvbtune** von *Dave Chapman*. Ich bevorzuge dieses Programm vor anderen Tuning-Programmen (z.B. **tuxzap** aus den DVB-Treiber-Sourcen), weil hier automatisch das Netzwerk-Interface der DVB-Karte gestartet werden kann. Bei anderen Lösungen muß noch zusätzlich ein "networkactivator"- Programm ausgeführt werden. Wenn man dvbtune ohne Parameter aufruft, erhält man eine Liste der zu verwendenden Parameter. Die folgenden Parameter werden für das Tuning in unserem Fall benötigt:

-cc [[00--33]]

Nummer der DVB-Karte; Es können bis zu 4 DVB-Devices angesprochen werden, falls mehr als eine Karte im PC enthalten sind. Im Normalfalle wird "-c 0" verwendet.

-ff ffrreeqq

Frequenz in Hz, auf die getuned werden soll

-pp [[HH,,VV]]

Polarisation, horizontal oder vertikal

-ss NN

Symbolrate in Hz

-nn mmpidd

Multicast-PID eventuell muß bei Verwendung eines DiSeq-Schalters noch der folgende Parameter gesetzt werden (erst ab Version 0.3 von dvbtune vorhanden):

-DD [[00--44]]

Stellung des DiSeq-Schalters Leider habe ich damit keine persönliche Erfahrung. Im Zweifelsfalle muß man die 5 möglichen Werte einfach mal durchprobieren. Welche Werte derzeit an dvbtune übergeben werden müssen, können Sie aus der folgenden Tabelle ablesen:

5.3.1 Tabelle der Parameter

Satellit	Astra 19.2°
Frequenz (Hz)	10773250
Polarisation	Horizontal
Symbolrate(S/s)	22000
Multicast-PID	251
IP-Radio-PID	252
Unicast-PID	253

Die IP-Radio-PID ist hier nur der Vollständigkeit halber aufgeführt. Für den Netzwerkzugang sind nur die Multicast-PID oder Unicast-PID von Bedeutung, je nach dem, ob man den Proxy-Zugang oder die VPN-Lösung verwenden möchte Bei mir sieht der Aufruf von **dvbtune** folgendermassen aus:

```
user@linux ~/ # dvbtune -c 0 -f 10773250 -p H -s 22000 -n 251
```

Ich spreche also den Satelliten auf Frequenz 10.733,250 MHz (-f 10773250) mit der ersten (und einzigen) DVB-Karte (-c 0) an. dvbtune gibt anschließend einige Werte zum Tuning-Vorgang aus, die allerdings - zumindest bei meiner TT budget-Karte - nicht besonders aussagekräftig sind. Wichtig ist, dass zum Schluß der Ausgabe folgendes steht:


```
FESTATUS: FEHASPOWER FEHASSIGNAL FEHASLOCK FEHASCARRIER  
FEHASVITERBI FEHASSYNC  
Successfully opened network device, please configure the dvb interface
```

Ist dies der Fall, hat das Tuning geklappt und das Netzwerk-Device der DVB- Karte kann konfiguriert werden.

5.4 Starten des Netzwerk-Interfaces

Die Treiber sind geladen, die DVB-Karte ist auf die richtige Frequenz getuned, jetzt sollte das Netzwerk-Interface zum Leben erweckt werden. Dies geschieht mit `ifconfig`. Zur Verdeutlichung stelle ich den Vorgang in drei Schritten dar, die allerdings auch in einem einzigen **ifconfig** zusammen ausgeführt werden können.

1. Schritt: Aktivieren von dvb00

```
user@linux ~/ # su -c "ifconfig -a dvb00"
```

2. Schritt: Zuweisen IP-Adresse und impliziter Start des Interface

```
user@linux ~/ # su -c "ifconfig dvb00 192.168.0.99"
```

Die hier vergebene IP-Adresse sollte natürlich noch von keinem Device in Ihrem Rechner bzw. Netzwerk benutzt werden und aus den privaten Nummernkreisen 192.168. oder 10.0.0. kommen. Die Adresse muß außerdem mit der Adresse des MULTICASTRECEIVEINTERFACE in der Datei `recv.ini` des Tellique-Clients übereinstimmen. Mehr dazu im Abschnitt 5.6. Die vergebene IP Adresse sollte sich zumindest im CLASS C vom restlichen Netzwerk unterscheiden, da sonst Routingprobleme auftreten können. Wenn also z.B. Ihre Rechner im Netzwerk den Adressbereich 192.168.1.1 - 192.168.1.254 benutzen (192.168.1.0 MASK 255.255.255.0), dann sollten sie dem dvb00 Device z.B. die Adresse 192.168.2.1 zuweisen. Empfehlen würde ich immer eine generelle Unterscheidung zwischen Routingadressen (ISDNROUTER, TDSL-DVB-Router usw.) und den Adressen des normalen Netzwerks.

3. Schritt: Zuweisen MAC-Adresse

```
user@linux ~/ # su -c "ifconfig dvb00 hw ether
```

00:01:02:03:04:05"

Die hier anzugebende MAC-Adresse sollte mit derjenigen übereinstimmen, die Sie bei Ihrer Registrierung (siehe Abschnitt 5.1) angegeben haben. Damit ist das Netzwerk-Interface dvb00 fertig konfiguriert und einsatzbereit. In `/var/log/messages` sollten Sie nun auch schon die ersten Ausgaben des Interfaces sehen. Nun kann getestet werden, ob auch tatsächlich Daten über das Interface empfangen werden.

5.5 Testen der Verbindung

Zum Testen, ob über das DVB-Interface tatsächlich Daten empfangen werden, läßt man sich einfach mittels `tcpdump` mal die eingehenden Datenpakete anzeigen. Dabei braucht man selbst gar keinen Traffic zu erzeugen, weil über das DVB-Netzwerk-Interface permanent Multicast-Datenpakete hereinkommen - vorausgesetzt, man hat die Karte auf die richtige Frequenz und die richtige Multicast-PID eingestellt. Über die gesetzten Multicast-Filter werden aus diesem permanenten Datenstrom nur diejenigen Pakete herausgefiltert, die

tatsächlich angefordert wurden. Mit

```
user@linux ~/ # su -c "tcpdump -ni dvb00"
```

leitet man den Datenstrom auf dem DVB-Interface auf die Standard-Ausgabe um. Nach kurzer Wartezeit sollten unaufhörlich UDP-Datenpakete auf dem Bildschirm angezeigt werden. Mit der Tastenkombination STRG + C können Sie `tcpdump` nach kurzer Zeit wieder abbrechen. Herzlichen Glückwunsch, der Datenempfang über den Satelliten funktioniert. Als Alternative zu `tcpdump` eignet sich auch das Tool `iptraf`. Dieses Tool von [Gerard Paul Java](#) hat gegenüber `tcpdump` den Vorteil, dass es auch den Datendurchsatz anzeigt und kann somit auch gut für spätere Performance-Messungen des SAT-Zugangs genutzt werden. Mit

```
user@linux ~/ # su -c "iptraf -d dvb00"
```

lassen Sie sich den Datenverkehr auf dem DVB-Netzwerkdevice anzeigen. Evtl. müssen Sie das Paket `iptraf` aber vorher noch aus Ihrer Distribution installieren.

5.6 Starten des Proxy

Nachdem nun die grundlegende Funktion des Datenempfangs soweit getestet ist, geht es in den nächsten Schritten um die Beschleunigung Ihres Internet-Zugriffs mit Hilfe des Tellique-Proxies. Vor dem ersten Start muß die Steuerdatei `recv.ini` geringfügig angepasst werden. Öffnen Sie dazu die Datei mit einem Editor Ihrer Wahl und suchen Sie den String `multicastreceiveinterface`. Entfernen Sie das evtl. vorhandene Kommentarzeichen (`#`) am Anfang der Zeile und setzen Sie die dort enthaltene IP-Adresse auf die von Ihnen bei der Einrichtung des DVB-Devices vergebene Adresse (siehe Abschnitt 5.4). Speichern Sie die Änderung und schliessen Sie die Datei. Nun können Sie den Proxy starten. Wechseln Sie dazu zunächst mit

```
user@linux ~/ # cd /opt/tellique
```

in das Verzeichnis, in dem sich das Programm befindet. Da der Proxy als Dämon ausgelegt ist, empfiehlt es sich, das Programm im Hintergrund auszuführen:

```
user@linux ~/ # su -c "./proxy &"
```

Der Proxy lauscht jetzt auf Port 9202 auf HTTP und FTP-Requests und auf Port 9203 für SOCKS. Das Programm schreibt außerdem eine Log-Datei `recv.log` im Programmverzeichnis, die man sich z.B. mit `tail -f` anschauen kann.

5.7 Einstellung Proxy am Browser

Als nächsten Schritt müssen Sie Ihrem Internet-Browser und/oder sonstigen Programmen mitteilen, dass der soeben gestartete Proxy für den Zugriff verwendet werden soll. Diese Einstellung hängt natürlich vom jeweils verwendeten Programm ab, daher kann ich hier keine vollständige Beschreibung vorlegen. Allerdings sollte die Handhabung immer ähnlich sein, dass für den Proxy eine IP-Adresse und ein Port angegeben wird. Für das HTTP- und FTP-Protokoll muß die Kombination `127.0.0.1:9202` angegeben werden, für SOCKS entsprechend `127.0.0.1:9203`. Im Folgenden habe ich eine Kurzanleitung für die drei unter Linux gebräuchlichsten Browser verfasst. Wer noch weitere Programme hier unbedingt aufgelistet haben möchte, soll mir die Anleitung, wie die Proxyeinstellung vorgenommen wird, zumailen.

5.7.1 Lynx

Da ich bislang nur Erfahrungen mit SuSE-Distributionen habe und ich nicht weiß, inwieweit die Systemkonfiguration über die Datei `/etc/rc.config` auch bei anderen Distributionen üblich ist, kann ich diese Anleitung nur auf SuSE bezogen liefern. Dort finden sich in besagter Datei zwei Proxy-Parameter, die wie folgt belegt werden müssen:

```
HTTPPROXY="http://127.0.0.1:9202/" FTPPROXY="http://127.0.0.1:9202/"
```

Darüber hinaus gibt es noch den Parameter "NOPROXY", unter dem eine Liste mit Hosts spezifiziert werden kann, die nicht über den Proxy sondern direkt angesprochen werden sollen (z.B. localhost, Rechner im internen Netzwerk). Nach einem Aufruf von `SuSEconfig` sind diese Werte aktiviert und **lynx** geht beim nächsten Internet-Zugriff über den Telique-Proxy. Für nicht-SuSE-User kann ich als kleinen Hinweis geben, dass man auch die entsprechenden Parameter mit `export` setzen kann. Temporär geht dies mit

```
user@linux ~/ # export httpproxy="http://127.0.0.1:9202/"
user@linux ~/ # export ftpproxy="http://127.0.0.1:9202/"
```

Wenn man danach **lynx** aufruft, sollte auch der Proxy verwendet werden.

5.7.2 Konqueror

Diese Beschreibung bezieht sich auf den **Konqueror 2.2.1** in der englischen Sprachversion. In der deutschen Sprachversion heißen die Menüpunkte sicherlich anders, dies sollte aber auch für jemanden, der im Englischen nicht ganz sattelfest ist, kein unlösbares Problem darstellen.

Aus der Menüleiste wählt man den Punkt "Settings" und im dortigen Pulldown-Menü den Punkt "Configure Konqueror...". Dadurch erhält man den Optionsbildschirm des Programms. Am linken Fensterrand ist eine Navigationsleiste, aus der man den Punkt "Proxies & Cache" auswählt. Im rechten Fensterbereich sollten Sie nun die Registerkarte "Proxies" angezeigt bekommen. Aktivieren Sie zunächst die Check-Box "Use Proxy", dadurch werden die Eingabefelder freigeschaltet. Geben Sie nun in den Feldern "HTTP Proxy:" und "FTP Proxy:" jeweils die IP-Adresse Ihres Loopback-Devices 127.0.0.1 an. Die drei "Port:"-Felder werden ebenfalls identisch mit Port 9202 gefüllt. Anschließend können Sie im Feld "No Proxy for" eine Liste von Rechnern und IP-Adressen angeben, die nicht über den Proxy geroutet werden sollen. Nachdem Sie alle Eingaben getätigt haben, drücken Sie entweder auf den "Apply"-Button oder auf "OK". Das war's.

5.7.3 Netscape/Mozilla

Die Konfiguration bei **Netscape** und **Mozilla** funktioniert in ähnlicher Weise wie beim **Konqueror**. Ich habe hier eine deutschsprachige Version des **Netscape 4.7** im Einsatz, weshalb nun die Menüpunkte in Deutsch beschrieben werden. In der Menüleiste wählt man den Punkt "Bearbeiten" und im Pulldown-Menü den Punkt "Einstellungen...". Es erscheint das Optionen-Fenster des Browsers. An der linken Fensterseite befindet sich eine Navigationsleiste, in der man zunächst den Punkt "Erweitert" expandieren muß (Click auf das "+" oder Doppelclick auf das Wort) und anschließend den Punkt "Proxies" anwählt. Im rechten Fensterbereich aktiviert man den Radio-Button "Manuelle Proxy-Konfiguration" und betätigt dann den "Anzeigen..."-Button. Im daraufhin angezeigten Fenster gibt man für alle angezeigten Internet-Protokolle als Adresse das Loopback-Device 127.0.0.1 an und als Port die Nummer 9202. Lediglich für "Socks:" wird der Port auf 9203 gesetzt. Anschließend bestätigen Sie die Eingaben in den Fenstern mit den "OK"-Button. Das war's.

5.8 Erster Login auf dem Proxy-Server

Der große Moment rückt näher. Rein technisch sind jetzt alle Voraussetzungen für die Nutzung von T-DSL via Satellit gegeben, es fehlt noch ein organisatorisches Detail: Die erste Anmeldung beim Proxy-Server. Der zuvor installierte und konfigurierte Tellique-Proxy ist ein Proxy-Client, der den Datenaustausch mit einem Proxy-Server durchführt. Um diesen Server nutzen zu können, müssen Sie sich einmalig bei diesem anmelden. Die Anmeldung funktioniert allerdings nur, wenn Sie zuvor die Registrierungsbestätigung der Deutschen Telekom (siehe Abschnitt 5.1) erhalten haben und den Proxy-Zugang in Ihren Registrierungsdaten angegeben haben. Für die erste Anmeldung starten Sie Ihren bevorzugten Internet-Browser (mit zuvor eingeschaltetem Proxy - siehe Abschnitt 5.7) und geben folgende Adresse ein: <http://127.0.0.1:2517/www/client/login/login.html>. Auf der erscheinenden Seite von ASTRA-NET werden Sie gebeten, Ihren Usernamen und das Passwort einzugeben - beides haben Sie von der Telekom zugeschickt bekommen, bzw. haben es bei der Registrierung selbst angepasst. Wenn Sie den Check-Button "Remember Password:" aktivieren, müssen Sie diesen Login nur dieses eine Mal durchführen und Nutzernamen und Passwort werden in der Datei recv.ini verschlüsselt gespeichert. Betätigen Sie jetzt den "Login"-Button und warten Sie auf die Bestätigung.

5.9 Herzlichen Glückwunsch!


Wenn Sie bis hierhin ohne Fehlermeldungen und Probleme gekommen sind, haben Sie es geschafft. Herzlichen Glückwunsch! :-). Suchen Sie sich im Internet eine richtig große Datei aus, die Sie schon immer mal haben wollten, und laden Sie diese herunter. Die Download-Geschwindigkeit sollte so um die 80-90 KByte/Sekunde betragen, wenn die Gegenseite ebenfalls entsprechend schnell ist.

6 Die Alternative:

6.1 Unterschiede zwischen VPN und Proxy

Bis zur Einführung von neuen VPN-Servern bei SES Astra Anfang August 2002 gab es einen entscheidenden Nachteil für den VPN-Zugang im Vergleich zu der im bisherigen Teil dieses HowTo's dargestellten Zugangsweise über den Tellique-Proxy: Beim VPN-Zugang wurde das Link-Sharing nicht eingesetzt. Dadurch wurde im Gegensatz zum Proxy die Telefonverbindung nur für den Hinweg der Daten in's Netz genutzt und alle Pakete kamen über den Satelliten zurück. Die Nichtberücksichtigung der Telefonleitung für den Rückkanal sorgte für subjektive Geschwindigkeitseinbußen beim Surfen und konnte sogar zu Übertragungsraten unterhalb der normalen Telefonverbindung führen. Da die Satelliten-Verbindung der einzige Übertragungsweg beim VPN war, wurde die Übertragungsgeschwindigkeit natürlich auch ausschließlich durch den Satelliten und dessen verfügbare Bandbreite definiert, während bei der Proxy-Lösung minimal die Performance der Telefonverbindung erreicht wurde, da die Pakete bei einem überlasteten Satelliten einfach über die terrestrische Verbindung zurückgeschickt wurden. Seit dem mit den neuen VPN-Servern bei SES Astra nun auch das Link-Sharing für den VPN-Zugang funktioniert, reduzieren sich die Unterschiede zwischen dem Proxy- und VPN-Zugang auf die Tatsache das der Proxy mit Multicasting arbeitet, während beim VPN die Datenpakete per Unicast versendet werden. Eine direkte Auswirkung für den täglichen Einsatz hat dieser Unterschied allerdings nicht. Gegenüber dem Proxy-Zugang hat der VPN-Zugang auch einige Vorteile. So wird mittels VPN jeglicher Datenverkehr mit dem Internet beschleunigt, während beim Proxy standardmässig nur das HTTP- und FTP-Protokoll beschleunigt wird. Desweiteren erhalten durch den Verzicht auf Multicasting auch Nutzer von vollwertigen DVB-s Karten (mit MPEG2-Decoder) die Möglichkeit einen stabilen Netzwerk-Zugang über den Satelliten zu verwirklichen. Außerdem ist der VPN-Zugang komplett mit Open-Source Software zu realisieren, während der Proxy nur als closed-source verfügbar ist und es zu diesem keine Open-Source Alternative gibt. Nach meinen bisherigen Erfahrungen gibt es keinen Unterschied beim Datendurchsatz zwischen dem Proxy und VPN. Die Übertragungsgeschwindigkeit variiert bei beiden Lösungen in Abhängigkeit von der aktuellen Belegung des Transponders.

6.2 Installation pptp-Client

Unter  <http://pptpclient.sourceforge.net/> befindet sich die Projektseite des pptp-Clients für Linux. Dieses Programm ist in der Lage über den Linux-pppd eine VPN-Verbindung zu einem Windows-VPN-Server aufzubauen. Der pppd muß dabei mindestens Version 2.4.0 oder höher sein, ansonsten muß man einen Patch dafür einspielen (Nähere Details im SAT-HowTo). Man kann dort das Ganze als vorkompiliertes RPM-Paket oder Source-Tarball herunterladen. Beide Varianten funktionierten bei meinen Versuchen einwandfrei. Nachdem man das RPM installiert bzw. die Source mit make kompiliert hat, kann man auch fast schon loslegen. Es fehlt allerdings noch ein Kernel-Modul, welches vor der Verwendung von pptp geladen werden muß. Das Modul heißt **ipgre.o** und befindet sich bei den Kernel-Modulen, normalerweise im Verzeichnis `/lib/modules/kernel/net/ipv4`. Dieses Modul wird (zumindest bei meiner Linux-Distribution) nicht automatisch geladen. Es muß daher entweder mit in den Kernel kompiliert werden oder als Modul manuell geladen werden. Letzteres geschieht am einfachsten mit `modprobe`, da auf diese Weise das Modul nur geladen wird, wenn dies vorher noch nicht geschehen ist. Das das Modul `ipgre` noch nicht geladen wurde, merkt man spätestens, wenn die VPN-Verbindung mit der folgenden Fehlermeldung abbricht:

```
Aug  4 22:49:34 athlon pptp[11434]: log[decapsgre:pptpgre.c:215]: short
read (4294967295): Protocol not available
Aug  4 22:49:34 athlon pppd[11433]: Terminating on signal 15.
```

Um diesen Fehler auszuschließen, empfiehlt es sich, beim automatisierten VPN-Verbindungsaufbau vor dem Aufruf von pptp ein `modprobe ipgre` einzubauen. Auf diese Weise kann man sicher sein, daß das Modul tatsächlich geladen ist.

6.3 Benutzererkennung in /etc/ppp/pap-secrets und /etc/ppp/chap-secrets

Um eine Verbindung zu einem VPN-Server aufzubauen, wird ein Benutzername und ein Passwort benötigt. Für das T-DSLvS-VPN handelt es sich hierbei um den Nutzernamen und das Passwort, welches Sie bei der Registrierung von der Telekom übermittelt bekamen bzw. und die von Ihnen entsprechend geänderten Angaben. Da der VPN-Tunnel im Grunde genommen nichts anderes als eine zusätzliche, virtuelle ppp-Verbindung mit dem VPN-Server als Einwahlserver darstellt, müssen die Angaben in /etc/ppp/pap-secrets und /etc/ppp/chap-secrets eingetragen werden. Fügen Sie zu den bestehenden Dateien einfach jeweils eine weitere Zeile im Format

`nutzernamen passwort`

ein. Da nur der Superuser eine Berechtigung für diese sensible Dateien hat, müssen natürlich auch alle Änderungen mit SU-Rechten durchgeführt werden. Beim Aufbau der VPN-Verbindung wird der pppd nach dem von Ihnen spezifizierten Benutzernamen entweder in pap-secrets oder chap-secrets suchen und das dort vorgefundene Passwort beim Verbindungsaufbau verschlüsselt an den VPN-Server übermitteln. Denken Sie bitte daran, dass der Zugang über VPN nur funktioniert, wenn sie diesen in Ihrem Registrierungsdaten angegeben haben. Ist dies nicht der Fall, wird der Verbindungsaufbau wegen unbekanntem Usernamen abgelehnt, auch wenn Sie in den Secret-Dateien vielleicht alles korrekt eingetragen haben.

6.4 Erster Verbindungstest

Nun kann man einen ersten Verbindungstest machen, ob der VPN-Tunnel zum Astra-Server auch korrekt aufgebaut wird. Der VPN-Server von TDSLvS heißt `ses.hsi.astra-net.com` (mehrere IP-Adressen im Subnetz 212.56.240.0). Ist dies geschehen, kann man mit dem Befehl

```
user@linux ~/ # pptp ses.hsi.astra-net.com debug user mru 1452 mtu 1452
```

prüfen, ob der Verbindungsaufbau funktioniert. Wer kein dial-on-demand benutzt, sollte natürlich zuvor die herkömmliche Verbindung in's Internet aufgebaut haben. Die Ausgabe von pptp sollte in etwa so aussehen:

```
Aug 21 07:47:00 athlon pptp[15982]:  
log[pptpdispatchctrlpacket:pptpctrl.c: 708]: Outgoing call established  
(call ID 0, peer's call ID 0).  
Aug 21 07:47:00 athlon pppd[15979]: pppd 2.4.1 started by root, uid 0  
Aug 21 07:47:00 athlon pppd[15979]: using channel 68  
Aug 21 07:47:00 athlon pppd[15979]: Using interface ppp0  
Aug 21 07:47:00 athlon pppd[15979]: Connect: ppp0 <--> /dev/pts/4  
Aug 21 07:47:00 athlon pppd[15979]: sent [LCP ConfReq id=0x1 <mru 1452>  
<asynctest 0x0> <magic 0x778161fe> <pcomp> <accomp>]  
Aug 21 07:47:02 athlon pppd[15979]: sent [LCP ConfReq id=0x1 <mru 1452>  
<asynctest 0x0> <magic 0x778161fe> <pcomp> <accomp>]  
Aug 21 07:47:02 athlon pppd[15979]: rcvd [LCP ConfAck id=0x1 <mru 1452>  
<asynctest 0x0> <magic 0x778161fe> <pcomp> <accomp>]  
Aug 21 07:47:02 athlon pppd[15979]: rcvd [LCP ConfAck id=0x1 <mru 1452>  
<asynctest 0x0> <magic 0x778161fe> <pcomp> <accomp>]  
Aug 21 07:47:03 athlon pppd[15979]: rcvd [LCP ConfReq id=0x1 <asynctest  
0x0> <auth chap m$oft> <magic 0xdcf8f6a8> <pcomp> <accomp>]  
Aug 21 07:47:03 athlon pppd[15979]: sent [LCP ConfAck id=0x1 <asynctest
```

```
0x0> <auth chap m$oft> <magic 0xdcf8f6a8> <pcomp> <accomp>]
Aug 21 07:47:03 athlon pppd[15979]: sent [LCP EchoReq id=0x0
magic=0x778161fe
Aug 21 07:47:03 athlon pppd[15979]: cbcplowerup
Aug 21 07:47:03 athlon pppd[15979]: want: 2
Aug 21 07:47:03 athlon pppd[15979]: rcvd [CHAP Challenge id=0x1
<b10fb56dc43aaa07>, name = "g2hsig04"]
Aug 21 07:47:03 athlon pppd[15979]: sent [CHAP Response id=0x1
<0000000000000000000000000000000000000000000000000000000dc6bb6118209d1f5b3295fbff525951
e2a5652cc217daab701>, name = "<username>"]
Aug 21 07:47:03 athlon pppd[15979]: rcvd [LCP EchoRep id=0x0
magic=0xdcf8f6a8]
Aug 21 07:47:04 athlon pppd[15979]: rcvd [CHAP Success id=0x1 "Welcome to
g2hsig04."]
Aug 21 07:47:04 athlon pppd[15979]: Remote message: Welcome to g2hsig04.
Aug 21 07:47:04 athlon pppd[15979]: sent [IPCP ConfReq id=0x1 <addr
192.168.97.2> <compress VJ 0f 01>]
Aug 21 07:47:04 athlon pppd[15979]: sent [CCP ConfReq id=0x1 <deflate 15>
<deflate(old#) 15> <bsd vl 15>]
Aug 21 07:47:04 athlon pppd[15979]: rcvd [IPCP ConfReq id=0x1 <addr
212.56.240.60> <compress VJ 0f 01>]
Aug 21 07:47:04 athlon pppd[15979]: sent [IPCP ConfAck id=0x1 <addr
212.56.240.60> <compress VJ 0f 01>]
Aug 21 07:47:04 athlon pppd[15979]: rcvd [CCP ConfReq id=0x1 <deflate 15>
<deflate(old#) 15> <bsd vl 15>]
Aug 21 07:47:04 athlon pppd[15979]: sent [CCP ConfAck id=0x1 <deflate 15>
<deflate(old#) 15> <bsd vl 15>]
Aug 21 07:47:04 athlon pppd[15979]: rcvd [IPCP ConfNak id=0x1 <addr
172.24.9.xxx>]
Aug 21 07:47:04 athlon pppd[15979]: sent [IPCP ConfReq id=0x2 <addr
172.24.9.xxx> <compress VJ 0f 01>]
Aug 21 07:47:04 athlon pppd[15979]: rcvd [CCP ConfAck id=0x1 <deflate 15>
<deflate(old#) 15> <bsd vl 15>]
Aug 21 07:47:04 athlon pppd[15979]: Deflate (15) compression enabled
Aug 21 07:47:06 athlon pppd[15979]: rcvd [IPCP ConfAck id=0x2 <addr
172.24.9.xxx> <compress VJ 0f 01>]
Aug 21 07:47:06 athlon pppd[15979]: local IP address 172.24.9.xxx
Aug 21 07:47:06 athlon pppd[15979]: remote IP address 212.56.240.60
Aug 21 07:47:06 athlon pppd[15979]: Script /etc/ppp/ip-up started (pid
15991)
Aug 21 07:47:06 athlon pppd[15979]: Script /etc/ppp/ip-up finished (pid
15991). status = 0x0
```

Wichtig ist vor allem die Angabe des mru/mtu-Wertes von 1452 (MRU = Max. Receiving Unit; MTU = Max. Transfer Unit, sprich die maximale Blockgröße für den Datenversand und -empfang). Standardmässig werden diese Parameter vom pppd nicht gesetzt und der Defaultwert von 1500 verwendet. Mit dieser Größe funktioniert jedoch der Datentransfer mit den VPN-Servern nicht. Sollte alles soweit korrekt gelaufen sein, weiß man schonmal, dass die technische Verbindung zum VPN-Server funktioniert und der VPN-Tunnel aufgebaut wird. Ein Blick auf das Routing (`route -n`) zeigt, dass ein zusätzliches ppp-Device (`ppp0`) mit einer Route zum VPN-Server aufgemacht wurde, die eigentliche Internet-Verbindung wird bei mir vom ISDN-Device `ipp0` aufgebaut, das VPN-Device ist dann `ppp0`:

Ziel	Router	Genmask	Flags	Metric	Ref	Use	Iface
------	--------	---------	-------	--------	-----	-----	-------

212.56.240.60	0.0.0.0	255.255.255.255	UH	0	0	0	ppp0
62.180.158.3	0.0.0.0	255.255.255.255	UH	0	0	0	ipp0
192.168.213.0	0.0.0.0	255.255.255.0	U	0	0	0	dvb00
192.168.97.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
0.0.0.0	62.180.158.3	0.0.0.0	UG	0	0	0	ipp0

Mit diesem Routing läuft aber noch kein Paket durch den Tunnel sondern es wird weiterhin die "normale" Leitung benutzt, weil die Default-Route auf das ISDN-Device zeigt und nicht auf den Tunnel. Außerdem muß das Tuning der DVB-Karte ja auch noch gegenüber dem Multicast/Proxy-Einsatz verändert werden.

6.5 Konfiguration DVB-Device

Zwischen der Konfiguration des DVB-Devices beim Proxy-Zugang und der VPN-Lösung gibt es nur einen Unterschied, und das ist die PID, die bei dvbtune, Parameter -n, angegeben werden muß. Folgen Sie also den Anweisungen in den Abschnitten 5.2 bis 5.4, statt der Multicast-PID 251 verwenden Sie nun aber die Unicast-Pid 253. Ich habe festgestellt, daß es nicht ausreicht, bei bestehendem Tuning für die Proxy-Lösung einfach dvbtune mit -n 253 nochmal aufzurufen um dann mit Unicast arbeiten zu können. Wenn man vorher über den Proxy gearbeitet hat, sollte man die dvb-Treiber erst neu laden und dann mit dvbtune auf Unicast tunen.

6.6 Verändern des Routings

Um nun auch die in's Internet zu versendenden Pakete tatsächlich durch den Tunnel zu schicken, muß die Default-Route vom eigentlichen Verbindungsdevice (bei mir ipp0) auf das VPN-Device (ppp0) gelegt werden. Dies alleine reicht aber noch nicht aus da dadurch auch die Encap-Pakete an den VPN-Server, die eigentlich über das ipp0-Device laufen sollten, über das VPN-Device geroutet werden, was so nicht funktioniert. Außerdem braucht man ja für die Nameserver-Anfragen gar keine hohe Übertragungsrate, sondern eher eine schnelle Reaktionszeit. Am Routing werden jetzt also folgende Veränderungen vorgenommen: Nameserver explizit über ISDN-Device routen (optional)

```
user@linux ~/ # su -c "route add dev ipp0"
```

Route zu VPN-Servern über ISDN-Device legen

```
user@linux ~/ # somebody@localhost:~ # su -c "route add
ses.hsi.astra-net.com dev ipp0"
```

Löschen der Default-Route

```
user@linux ~/ # su -c "route del default"
```

neue Default-Route auf VPN-Device

```
user@linux ~/ # su -c "route add default dev ppp0"
```

Die Änderungen des Routings müssen natürlich bei bestehender Internet-und VPN-Verbindung durchgeführt werden, da ansonsten z.B. das Device ppp0 gar nicht verfügbar ist. Das neue Routing sieht dann in etwa so aus:

--	--	--	--	--	--	--	--

Ziel	Router	Genmask	Flags	Metric	Ref	Use	Iface
212.56.240.60	0.0.0.0	255.255.255.255	UH	0	0	0	ppp0 (VPN)
62.134.11.4	0.0.0.0	255.255.255.255	UH	0	0	0	ippp0
62.180.158.1	0.0.0.0	255.255.255.255	UH	0	0	0	ippp0 (DNS)
195.182.110.132	0.0.0.0	255.255.255.255	UH	0	0	0	ippp0 (DNS)
192.168.213.0	0.0.0.0	255.255.255.0	U	0	0	0	dvb00
192.168.97.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
212.56.240.62	0.0.0.0	255.255.255.255	UH	0	0	0	ippp0 (VPN)
0.0.0.0 (Default)	0.0.0.0	0.0.0.0	U	0	0	0	ppp0

6.7 Zweiter Test

Der bislang an dieser Stelle dargestellte Test mittels Ping auf einen externen Rechner funktioniert aufgrund des Link-Sharings nicht mehr. Die beim Ping versendeten Datenmengen sind in der Regel zu klein, um tatsächlich über den Satelliten geroutet zu werden. Statt dessen sollten Sie versuchen per ftp oder wget eine größere Datei aus dem Internet zu laden und dabei die Performance und Datenpakete auf den beteiligten Devices mittels tcpdump oder iptraf kontrollieren. Wenn dies bei Ihnen funktioniert, haben Sie Ihren VPN-Zugang über T-DSL via Satellit erfolgreich eingerichtet. Herzlichen Glückwunsch.

7 Tips, Tricks, Bugs und was sonst noch so dazu gehört

7.1 Automatisierung

Nachdem für den ersten Zugang nun alles soweit funktioniert hat, soll das System nun dahingehend konfiguriert werden, dass der T-DSL via Satellit-Dienst auch im täglichen Gebrauch verwendet werden kann. Dabei ist zu beachten, dass die auf der vorherigen Seite beschriebenen Aktionen vom Laden der Treiber bis zum Starten des Proxy (Abschnitte 5.2 bis 5.6) bzw. dem Aufbau des VPN-Tunnels (siehe Kapitel 6) vor der Benutzung durchgeführt werden müssen. Der Verbindungstests aus Abschnitt 5.5 bzw. Abschnitt 6.7 kann dabei natürlich ausgelassen werden. Grundsätzlich gibt es zwei Möglichkeiten, wo diese Schritte ausgeführt werden können und es hängt stark von der persönlichen Handhabung des Systems und der Verwendung der DVB-Karte ab, welche der Alternativen nun bevorzugt werden sollte.

7.1.1 Initialisierung beim Wechsel des Runlevels

Wenn die Einwahl des Rechners in's Internet mit dial on demand automatisch abläuft und die DVB-Karte weitestgehend nur für den Internet-Zugang und nicht für Digital-TV oder -Radio genutzt wird, ist es am einfachsten, die komplette Initialisierung des Satelliten-Zugangs beim Wechsel in einen Netzwerk-Runlevel (z.B. init 3 oder init 5) durchzuführen. Der Vorteil dabei ist, daß die Schritte nur einmalig durchgeführt werden müssen und danach der Zugang permanent zur Verfügung steht. Beim Verlassen des entsprechenden Runlevels sollte die Initialisierung natürlich wieder rückgängig gemacht werden. Da die Vorgehensweise, wie Programme beim Wechsel eines Runlevels gestartet bzw. gestoppt werden, von Distribution zu Distribution unterschiedlich gehandhabt wird, kann ich hier keine pauschale Anleitung dazu geben. Minimal sollte es jedoch genügen, in einem Skript die beschriebene Befehlsfolge zu codieren und dieses Skript mit einem symbolischen Link in das entsprechende Verzeichnis für den Runlevel einzutragen. Näheres finden Sie in der Dokumentation Ihrer Distribution. In Zusammenarbeit mit *Dirk Wellmann* eine komplette Start/Stop-Prozedur entwickelt, die - wie beschrieben - die Initialisierung und Deinitialisierung des Satellitenzugangs beim Wechsel in die Netzwerk-Runlevel vornimmt. Ausgerichtet ist diese auf die Version 7.3 der SuSE-Distribution, sollte aber durch einige Änderungen von Markus Dahlweid auch unter SuSE 8.0 und anderen Distributionen lauffähig sein. Die beteiligten Skripts sowie einige Dokumentation zur Installation können hier heruntergeladen werden.

7.1.2 Initialisierung beim Aufbau der Internet-Verbindung

Für User, die ihre DVB-Karte neben dem Netzwerk-Zugriff auch häufig für Digital-TV und -Radio benutzen, ist der statische und monolithische Aufbau des im vorherigen Punkt beschriebenen Initialisierungssystems nicht uneingeschränkt geeignet. Zumindest das Tuning der Karte wird häufig verändert und muß vor jeden Netzwerk-Zugriff geprüft werden. Für diese User empfiehlt es sich, den Initialisierungsprozess der DVB-Karte beim Aufbau der Internet-Verbindung durchzuführen. Auf diese Weise wird sichergestellt, dass das Tuning der Karte korrekt ist und alle benötigten Module geladen sind. Eine Möglichkeit hierfür wäre, in das ip-up-Skript, welches sich normalerweise im Verzeichnis /etc/ppp befindet, die entsprechenden Befehle einzubauen. Ob hier tatsächlich der gesamte DVB-Start-Ablauf nachvollzogen werden muß, hängt von den persönlichen Vorlieben ab. Ich würde dazu tendieren, die DVB-Treiber beim Systemstart zu laden um diese permanent zur Verfügung zu haben, weil ohne die Treiber auch andere Nutzungsarten der DVB-Karte nicht möglich sind. Zumindest der Aufruf von `dvbtune` und `ifconfig dvb00` sollte aber in diesem Skript vorgenommen werden. Sofern kein Dial-on-demand verwendet wird, kann auch der Start des Proxy hier geschehen, weil dieser nur bei bestehender Netzwerk-Verbindung benötigt wird. Mit dial-on-demand muß der Proxy allerdings schon vor der Netzwerkverbindung vorhanden sein, weil der Proxy den Verbindungsaufbau triggert. Um bei diesem Punkt ein wenig Unterstützung zu bieten, habe ich die im vorherigen Abschnitt genannte Start-Prozedur so modular gestaltet, dass jeder Bestandteil des DVB-IP-Systems mit einem eigenen Skript einzeln gestartet und gestoppt

werden kann. Insofern kann also der Tarball mit den Initialisierungs-Skripten aus Abschnitt 7.1.1 auch hier von Nutzen sein.

7.1.3 Anpassungen /etc/ip-up Script für VPN-Zugang

Grundsätzlich gelten die in den vorherigen beiden Abschnitten skizzierten Überlegungen in gleicher Weise für den Proxy-Zugang als auch den VPN-Zugang. Einziger Unterschied ist, dass der Proxy nicht gestartet wird, da an dessen Stelle der Aufbau des VPN-Tunnels tritt. Dieser kann nur bei bestehender Internet-Verbindung eingerichtet werden, weshalb es hier keine Alternative zur Einbindung des VPN in das ip-up Script gibt. Aufgrund mangelnder Erfahrung mit anderen Distributionen kann ich hierbei auch wieder nur von der Vorgehensweise bei SuSE-Linux ausgehen und hoffe, dass die Strategien zum Aufbau einer Wählverbindung nicht allzusehr variieren. Unter SuSE wird zum Aufbau der Internet-Verbindung über pppd oder ipppd (ISDN) das Script **/etc/ip-up** verwendet. In diesem ist bei SuSE schon vorgesehen, daß ein weiteres Script **/etc/ip-up.local** ausgeführt wird, sofern dieses vorhanden und ausführbar ist. In diesem Script sollen die User-anhängigen Besonderheiten beim Verbindungsaufbau eingetragen werden, da das eigentliche ip-up-Script beim Update der Distribution evtl. überschrieben wird. Somit eignet sich das ip-up.local Script hervorragend für den Aufbau des VPN-Tunnels und das Anpassen des Routings. Dabei kann man sich zu Nutze machen, dass ip-up (und somit auch ip-up.local) bei der VPN-Verbindung zweimal ausgeführt wird - einmal beim Aufbau der Internet-Verbindung über das primäre ppp-Device (bei mir ippp0) und zum zweiten Mal beim Aufruf von pptp mit dem VPN ppp-Device (bei mir ppp0). Dabei wird jeweils das zu verbindende Interface als Parameter mitgegeben und man kann über eine Case-Anweisung steuern, wann welche Befehle ausgeführt werden. Ich habe ein Beispiel eines ip-up.local Scripts für den Aufbau der VPN-Verbindung in den Tarball mit den TDSLvS-Startskripten hineingepackt. Desweiteren ist das System nun auch soweit flexibilisiert, dass die Skripte über einen Parameter gesteuert entweder den Proxy-Zugang oder den VPN-Zugang initialisieren. Da ich nicht weiß, inwieweit diese Lösung SuSE-spezifisch ist, kann ich keine Garantie dafür übernehmen, dass dies bei anderen Distributionen auch funktioniert, aber zumindest einen Anhaltspunkt, wie's gemacht wird, sollte dies geben können. Lesen Sie dazu bitte auch die README-Dateien in dem Script-Tarball.

7.2 Zugang mit ISDN - Dial on Demand

Es ist grundsätzlich möglich, Linux so zu konfigurieren, dass die Einwahl in's Internet automatisch erfolgt, sobald ein externer Rechner angesprochen wird. Dieses Vorgehen nennt sich "Dial on Demand (DoD)" und ist vor allem für ISDN-Verbindungen geeignet, da hier der Verbindungsaufbau sehr schnell geht. Prinzipiell läuft DoD so ab, daß im Routing eine Defaultroute festgelegt wird, die auf das ISDN-Device zeigt und somit alle Requests an IP-Adressen, die nicht in eine der vorstehenden Routen passen, einen Verbindungsaufbau über das ISDN-Device generieren. Sollten über einen definierbaren Zeitraum dann keine Datenpakete mehr transferiert werden, wird die Verbindung auch automatisch wieder abgebaut. Ein solches Vorgehen hat natürlich einen hohen Bequemlichkeitslevel, weil man ja quasi eine virtuelle Standleitung zur Verfügung hat. Allerdings muß das System auch äußerst genau eingerichtet sein, damit nicht durch Hintergrundprozesse oder Ähnliches unnötige und kostenpflichtige Verbindungen aufgebaut werden. Nähere Infos zu diesem Thema gibt es im ISDN-HowTo von Klaus Franken. Leider stellt der TelliQue-Proxy bei eingeschaltetem DoD in seiner Default-Einstellung alle paar Minuten eine Verbindung in's Internet her. Dies liegt an dem Announcement-Channel, auf dem Daten asynchron über den Satelliten übertragen werden können. Diese Funktionalität ist derzeit unter Linux jedoch nicht nutzbar, daher kann man diesen Dienst getrost abstellen. Am einfachsten umzusetzen ist dies durch eine Änderung der Datei license.ini, die sich im Verzeichnis des TelliQue-Proxy befindet. Dazu öffnet man die besagte Datei mit dem persönlichen Lieblings-Editor und ändert die Zeile

```
Client/TelliCast=on
```

```
in
```

Client/TelliCast=off

Nach dem Neustart des Proxy werden keine eigenständigen Verbindungen mehr durch den Proxy aufgebaut. Ein weiterer Punkt im Zusammenspiel zwischen ISDN/DoD und T-DSL via Satellit wurde schon im vorhergehenden Abschnitt 7.1.2 angesprochen: Der Proxy muß permanent geladen sein, damit die HTTP- und FTP-Requests überhaupt zu einem Verbindungsaufbau führen. Ist der Proxy nicht geladen, erhält das Anwendungsprogramm eine Fehlermeldung. Für den VPN-Zugang treffen diese beiden proxy-spezifischen Punkte nicht zu, dafür gibt es dort aber andere DoD-relevanten Probleme. So dauert der Verbindungsaufbau etwas länger als gewohnt, weil neben der primären Wählverbindung ja noch die zweite ppp-Verbindung für den VPN-Tunnel etabliert werden muß. Dies kann sich vor allem dann negativ bemerkbar machen, wenn man einen relativ kurzen Timeout definiert hat, nachdem die Verbindung wieder abgebaut werden soll und somit innerhalb einer Surf-Session des öfteren die Verbindung aufgebaut und wieder gekappt wird. In diesem Falle hilft es nur, den Timeout heraufzusetzen oder aber die Verzögerungen in Kauf zu nehmen. Desweiteren werden die Pakete des Requests, der den Verbindungsaufbau initiiert, gar nicht über den Satelliten geroutet und somit auch nicht beschleunigt. Grund hierfür ist, dass beim Versand des Requests die Default-Route noch auf das primäre ppp-Device zeigte und das Routing für die VPN-Verbindung erst im nachhinein verändert wurde. Der erste Request bekommt von dieser Änderung nichts mehr mit und wird über das primäre Device beantwortet. Sofern dieser erste Request lediglich der Aufruf einer Internet-Seite oder ein DNS-Request war, ist die Sache noch zu verschmerzen. Wer jedoch sofort mit einem FTP-Download einer großen Datei die Verbindung aufbaut, wird über den gesamten Download-Vorgang nicht über die Geschwindigkeit der normalen Wählverbindung hinauskommen. Abhilfe schafft hier ein vorgeschalteter Ping, etwa in der Art:

```
user@linux ~/ # ping -c 5 x.y.z && wget url.zu.grosser.datei
```

Ich persönlich habe mein System mit ISDN/DoD konfiguriert und habe bis auf die geschilderten Kleinigkeiten weder mit dem Proxy noch mit dem VPN irgendwelche negative Erfahrungen damit gemacht.

7.3 Einsatz im Netzwerk

Viele Linux-Nutzer haben nicht nur einen Computer, sondern gleich ein kleines oder größeres Netzwerk aufgebaut, bei dem ein Rechner die Funktion eines Routers in's Internet übernimmt und für den Verbindungsaufbau sorgt. Für diese Konstellationen ist es natürlich wünschenswert, auch den Client-Rechnern im Netzwerk den Vorteil der Satellitenbeschleunigung zu gewähren. Unter Verwendung des Telli-Proxy und solange die DVB-Karte und das Verbindungsgerät (ISDN-Karte, Modem o.ä.) für die herkömmliche Internet-Verbindung in einem Rechner stecken, ist dies relativ einfach möglich, indem man bei den Anwendungsprogrammen (Browser, FTP-Client, Filesharing-Clients) entsprechend der im Abschnitt 5.7 aufgezeigten Vorgehensweise die Proxy-Einstellungen einträgt. Dabei muß allerdings für alle Clients die Loopback-Adresse 127.0.0.1 durch die IP-Adresse des Routers ersetzt werden. Problematisch wird die Sache erst, wenn die DVB-Karte in einem anderen Rechner steckt, als demjenigen, über den die Internet-Verbindung aufgebaut wird - vor allem in Verbindung mit ISDN Dial-on-Demand und kurzen Timeouts. Über den Telli-Proxy sieht der Verbindungsaufbau mit DoD und Satellitenzugriff normalerweise so aus, dass bei einer Anforderung von Daten eines externen Rechners das erste Datenpaket über den Proxy die ISDN-Verbindung herstellt, der Telli-Proxy die Verbindung mit den Proxy-Servern aufnimmt und eine IP-Adresse aushandelt, anhand deren der Multicast-Filter gesetzt wird. Danach können die Daten mit Satellitenbeschleunigung empfangen werden. Der Multicast-Filter wird wieder entfernt, wenn entweder über einen längeren Zeitraum keine Datenpakete mehr empfangen werden oder die ISDN-Verbindung abgebaut wird. Wenn die ISDN-Verbindung über einen anderen Rechner aufgebaut wird, und diese aufgrund eines sehr kurzen Timeout-Wertes (≤ 30 Sekunden) vor dem automatischen Abbau des MC-Filters wegen Inaktivität wieder beendet wird, bleibt der gesetzte MC-Filter bestehen. Beim nächsten Internet-Zugriff verhindert der noch bestehende Filter dann, dass ein neuer MC-Filter gesetzt wird. Da sich aber die ausgehandelte IP-Adresse zwischenzeitlich geändert hat, gehen die angeforderten Daten in's Leere und nach einer gewissen Zeit meldet der

Proxy-Client ein "Bad Gateway". "Glücklicherweise" wird dabei dann der falsche MC-Filter entfernt, weshalb der nächste Zugriff dann wieder erfolgreich ist. Der bevorzugte Workaround für dieses Problem wäre, beide Devices (DVB- und ISDN-Karte) in einen Rechner zu stecken. Wenn dieses aber aus räumlichen Gründen nicht möglich ist, bleibt noch die Möglichkeit, den Verbindungsaufbau manuell zu triggern und auch wieder abzubauen, allerdings darf letzteres erst geschehen, wenn der MC-Filter automatisch entfernt wurde. Denkbar wäre es auch, im `/etc/ppp/ip-down` Skript einen entsprechenden Aufruf einzubauen, der remote auf dem Rechner mit dem DVB-Device den Multicast-Filter entfernt. Dies ist allerdings eine Aufgabe für Scripting- und Netzwerk-Spezialisten und übersteigt meine Kenntnisse auf diesen Gebieten. Ich wollte hier nur auf die theoretische Möglichkeit hinweisen. Für den VPN-Zugang spielt es allerdings keine Rolle, ob DVB- und ppp-Device in einem oder mehreren Rechner stecken. Entscheidend für Auf- und Abbau der Verbindung ist einzig das ppp-Device. Allerdings dürfte das Routing etwas komplizierter ausfallen, wenn der Verbindungsrechner nicht mit dem DVB-Rechner identisch ist. Für den Einsatz des VPN-Zugangs in einem Netzwerk empfiehlt es sich, zusätzlich einen Proxy auf dem Verbindungsrechner aufzusetzen, den man in ähnlicher Weise wie den Tellico-Proxy in den Browsern der Client-Maschinen angibt. Jeder beliebige Proxy ist dazu geeignet (z.B. squid). Theoretisch bestünde auch noch die Möglichkeit, ohne Proxy mit Masquerading die Verbindung der Clients in's Internet zu realisieren, allerdings bin ich mir nicht sicher, ob dies tatsächlich funktioniert. Ich habe den VPN-Zugang vom Client aus lediglich über squid ausprobiert und es funktionierte hervorragend.


7.4 Bugs und Einschränkungen

Während der Pilotphase des Dienstes haben sich einige Schwachstellen der momentanen Implementierung unter Linux gezeigt, die ganz unterschiedliche Gründe haben. Ich habe hier alle Probleme und Einschränkungen aufgelistet, von denen ich bis zum jetzigen Zeitpunkt Kenntnis erhalten habe und versucht, deren momentanen Status zu dokumentieren. Die Liste erhebt keinen Anspruch auf Vollständigkeit oder permanente Aktualität.

Systemstillstände oder Reboots bei DVB-s Karten

Es scheint noch ein Problem in den Treibern mit dem Multicast-Support für die vollwertigen DVB-s Karten mit MPEG2-Decoder on board zu geben. Mehrere Teilnehmer des Pilotprojekts haben berichtet, dass das System entweder einfach stehen bleibt oder selbstständig einen Reboot durchführt. Mit den budget-Karten (ohne MPEG-Decoder) ist dieses Verhalten noch nicht aufgetaucht. Außerdem beschränkt sich das Problem nur auf die Proxy-Lösung, der VPN-Zugang funktioniert einwandfrei.

Status: unklar. Soweit ich weiß, wird derzeit das Problem in Zusammenarbeit mit den Treiberentwicklern analysiert.

Lösung/Workaround: Statt des Tellico-Proxy den VPN-Zugang verwenden. Statt einer DVB-s Karte eine budget-Karte (TT Budget PCI oder WinTV nova) einsetzen. ;-) Aktuelle Entwicklung verfolgen und immer neueste Treiberversionen verwenden. Infos zu dem Thema gibts im Webforum von  www.ipviasky.net oder in der Linux-DVB-Mailingliste.

periodischer Verbindungsaufbau durch den Proxy

Dieses Problem ist schon im Abschnitt 7.2 beschrieben worden. Der Verbindungsaufbau wird bei aktiviertem Announcement-Channel durchgeführt.

Status: gelöst

Lösung/Workaround: siehe Abschnitt 7.2

"Bad Gateway"-Meldung bei jedem zweiten Verbindungsversuch

Dieses Problem ist ebenfalls schon im Abschnitt 7.3 beschrieben worden. Bei der Verwendung von ISDN/DoD und Einwahl in's Internet über einen anderen Rechner sowie sehr kurzer Timeout-Einstellung, wird der Multicast-Filter nicht wieder abgebaut und verhindert den Datenempfang beim nächsten Verbindungsversuch.

Status: gelöst

Lösung/Workaround: siehe Abschnitt 7.3

Probleme bei Nutzung von wget

Carsten Koch hat bei seinen Tests mit dem Tellique-Proxy festgestellt, dass dieser nicht 100% transparent ist. Bei der Nutzung von wget zum Download eines ganzen FTP-Verzeichnisses verändert der Proxy wohl das Verzeichnislisting so, dass wget damit nichts mehr anfangen kann. Nähere Infos dazu gibt's im Webforum von IPviaSky.net im Thread Tellique Proxy 1.0 funktioniert nicht zusammen mit wget. Ein weiteres Problem in diesem Zusammenhang scheint die Tatsache zu sein, dass der Tellique-Proxy die Timestamps der übertragenen Dateien nicht mitliefert. Dadurch ist kein echtes Mirroring möglich, bei dem nur veränderte Dateien übertragen werden sollen und die Timestamps auf Original und Mirror identisch sein sollten.

Status: das Problem wurde an die Entwickler von Tellique weitergeleitet. Bislang ist noch keine Stellungnahme dazu publik geworden.

Lösung/Workaround: Statt des Tellique-Proxy den VPN-Zugang verwenden. Warten auf neue Proxy-Version... ;-)

Fehlermeldung "unresolved symbol kmappagetabable" beim Laden der Treiber

Beim Laden der Treiber erscheint oben stehende Fehlermeldung. Vermutlich verwenden Sie einen erweiterten Kernel mit "User-highmem-Support".

Status: gelöst

Lösung/Workaround: Fügen Sie in die Datei saa7146core.c im DVB-Treiber-Verzeichnis die folgende Zeile unmittelbar unter der Anweisung "#include <linux/vmalloc.h>" ein: #include <linux/highmem.h> / fix for the unresolved symbol / Danke an Dirk Wellmann für den Hinweis.

Alles funktioniert, aber keine Beschleunigung

Sie haben alles korrekt konfiguriert und nach Anleitung durchgeführt. Sie erhalten keinerlei Fehlermeldungen und dennoch bleibt die Download-Geschwindigkeit sogar weit hinter ISDN-Niveau zurück. Dieses Problem hatte ich nach der Pilotphase auch, als der Regelbetrieb über neue Proxyserver lief (212.56.240.36 - .39). Über die alten Pilotserver mit anderen IP-Adressen funktionierte die Beschleunigung wunderbar, aber mit den neuen Servern wurde der Datentransfer sogar eher noch gebremst als beschleunigt. Der Grund dafür war bei mir die SuSE-personalFirewall aus der Distribution SuSE 7.3. In dieser Firewall muß irgendeine Regel enthalten sein, die dafür sorgte, daß die Datenpakete von den neuen Servern kommentarlos gedropt wurden. Nachdem ich das Problem lokalisiert hatte, stellte ich fest, dass es auch nicht ausreicht, die Firewall weiter zu öffnen, sondern mußte Sie komplett deinstallieren. Dirk Vornheder berichtete mir kürzlich auch davon, dass er die gleichen Erfahrungen auch mit der SuSEfirewall2 aus der SuSE 8.0-Distribution gemacht hat.

Status: gelöst

Lösung/Workaround: Vermeiden Sie die Benutzung der SuSE-personalFirewall und der SuSEfirewall2. Nutzen Sie statt dessen eigene iptables-Lösungen oder andere Firewalls, die Ihnen zumindest mitteilen, dass Pakete gedropt werden, damit man einen Anhaltspunkt zur Recherche hat.

Keine Beschleunigung beim EMail-Verkehr

Wer darauf gehofft hat, nun auch beim Abholen seiner umfangreichen EMail eine schnellere Performance zu erhalten, wird zunächst enttäuscht sein. Die Beschleunigung über den Satelliten bezieht sich nur auf den HTTP- und FTP-Transfer. IMAP, POP3 unterstützen keine Proxy-Verbindungen und kommen daher für eine Beschleunigung ohne weiteres nicht in Frage. Implizit gilt diese Einschränkung für alle Programme/Dienste, die keine Proxy-Unterstützung haben.

Status: works as designed. Es gibt aber Umgehungsmöglichkeiten

Lösung/Workaround: - T-DSL via Satellit über alternative VPN-Verbindung nutzen

- Webmailer benutzen, der Übertragung per HTTP durchführt

- MUA/MTA entwickeln, der Proxy-Support hat. ;-)

- Port-Forwarding in der rcv.ini des Tellico-Proxies angeben. In einem FAQ-Beitrag auf www.ipviasky.com wird beschrieben, welche Einträge man dafür in der rcv.ini des Proxy vornehmen muß. Ich selbst habe diese Alternative noch nicht getestet und kann daher nicht sagen, ob sie funktioniert. Sie haben den VPN-Zugang eingerichtet und nutzen Dial-on-Demand für Ihre Wahlverbindung. Beim ersten Zugriff auf das Internet erhalten Sie keine Beschleunigung, obwohl der VPN-Tunnel korrekt aufgebaut wird. Nachfolgende Requests werden allerdings voll beschleunigt. Der Grund liegt im Routing zum Zeitpunkt des ersten Requests, der den Aufbau der Wahlverbindung triggert. Zu diesem Zeitpunkt liegt die Default-Route noch auf dem primären ppp-Device und wird erst später auf den VPN-Tunnel verlegt. Der erste Request bekommt von dieser Änderung jedoch nichts mehr mit und erhält seine Antwort noch über das primäre ppp-Device, wodurch natürlich eine Beschleunigung nicht möglich ist. (siehe auch Abschnitt 7.1.3)

Status: works as designed

Lösung/Workaround: Solange der Trigger-Request kein ftp-Download einer großen Datei ist, sollte sich das Problem nicht allzusehr bemerkbar machen. Für den FTP-Fall hilft es, ein paar pings vor den Download zu schalten, damit der ftp erst startet, wenn der VPN-Tunnel und das Routing komplett aufgebaut sind. Der VPN-Tunnel wird wie gewünscht aufgebaut, allerdings bricht die Verbindung ab, sobald die ersten Pakete durch den Tunnel gehen sollen. In /var/log/messages findet sich eine Fehlermeldung in folgender Art:

/var/log/messages	
Aug 4 22:49:34 athlon pptp[11434]: log[decapsgre:ptpgre.c:215]: short read (4294967295):	Protocol not available
Aug 4 22:49:34 athlon pppd[11433]: Terminating on signal 15.	

Dieser Abbruch ist auf das Fehlen des Kernel-Moduls ipgre.o zurückzuführen. Laden Sie das Modul mit modprobe nach und das Problem tritt nicht mehr auf.

Status: gelöst

Lösung/Workaround: Laden sie das Modul **ipgre.o** vor dem Aufbau der VPN-Verbindung mit **modprobe ipgre**. Nähere Informationen dazu befinden sich im Abschnitt 6.2

Der VPN-Tunnel wird wie gewünscht aufgebaut, allerdings bekommen Sie keine Rückmeldung auf

Anforderungen, die durch den Tunnel gehen. Eventuell kann es sogar so aussehen, dass einige Internetseiten erreichbar sind, während die Verbindung zu anderen Sites nicht funktioniert. Vermutlich ist die Angabe der MRU und MTU-Werte nicht vorhanden oder falsch gesetzt. Beide Parameter sollten auf den Wert 1452 gesetzt werden. Dies kann entweder direkt beim Aufruf von pptp geschehen oder aber unter `/etc/ppp/options` eingetragen werden.

Status: gelöst

Lösung/Workaround: Geben Sie den korrekten Wert von mru und mtu an (1452). Nähere Informationen hierzu unter Abschnitt 6.4

Werden USB-DDVB-Geräte unter Linux unterstützt

Die momentan aktuellen DVB-Treiber für Siemens unterstützen nur die internen PCI-DVB-Karten und keine externen USB-Geräte. Zwar wird in den Treibern am USB-Support gearbeitet, aber dieser steckt noch in einem sehr frühen Stadium und ist noch weit entfernt von der Routinetauglichkeit. Nähere Informationen erhält man in der Linux-DVB-Mailingliste. Leider scheint die Entwicklung der USB-Treiber nicht mehr voranzukommen, da es an Unterstützung der Hardware-Hersteller mangelt. Hier ein Statement eines Convergence-Mitarbeiters in der Linux-DVB-Mailingliste:

We don't have any documentation and no technical support neither from Hauppauge nor Technotrend (they made the initially design). Right now I don't have much motivation to continue reverse engineering. The protocol is basically completely known, except the {0x27, 2, ?, ?} sequence which is still missing. Contact me if anybody of you wants to continue working, then I'll try to explain anything you need to know and help wherever possibly.

Status: in Arbeit. Ob und wann der USB-Support in den DVB-Treibern serienreif sein wird, läßt sich derzeit nicht sagen.

Lösung/Workaround: - Linux-DVB-Mailingliste verfolgen und neueste Treiberversionen verwenden

- interne PCI-DVB-Karte verwenden

7.5 Ausblick

Sie sind am Ende meines HowTo angelangt und sollten jetzt so viele Informationen an der Hand haben, dass Sie den Zugang zu T-DSL via Satellit auf Ihrem Linux-System konfigurieren können. Ich würde mich freuen, Ihnen dabei tatsächlich geholfen zu haben und möchte Sie nochmals auffordern, mir Kritikpunkte und Verbesserungsvorschläge zu diesem Dokument zukommen zu lassen. Natürlich ist dieses Dokument noch nicht komplett, es gibt noch einige Punkte, die ich mir für die Zukunft vorgenommen habe. Dabei geht es allerdings schon mehr um die Expertenfragen und nicht mehr um die grundsätzliche Funktionsweise und Konfiguration. Für alle, die es interessiert, hier meine Ideen für weitere Abschnitte dieser Anleitung:

Sicherheitsaspekte:

Firewalling und DVB

Hierzu gibt es bislang noch gar keine Informationen. Wie muß die Firewall auf dem Rechner konfiguriert werden? Welche Ports müssen offen bleiben? Können Hacker über den Satelliten an meinen Rechner?

Setup mit alter Treibergeneration

Die gesamte Dokumentation bezieht sich auf die Treibergeneration ab Version 0.9.x, die nur in Verbindung mit Kernen ab 2.4.x funktioniert. Multicast-Networking war allerdings schon mit den älteren Treibern und Kernen

möglich, läuft aber etwas anders ab. Zwar gibt es die Anleitung, wie man die Treiber konfigurieren muß, schon in dem allgemeinen SAT-HowTo, allerdings nur in englisch und nicht explizit für den T-DSL via Sat-Zugang.


Übersetzung in's Englische

Englisch ist die Sprache für Dokumentationen unter Linux. Außerdem ist der Zugang zu T-DSL via Satellit nicht nur auf den deutschsprachigen Raum beschränkt, sondern kann europaweit empfangen werden. Daher halte ich eine Übersetzung dieses Dokumentes in's Englische für notwendig. Soweit meine Liste mit offenen Punkten, die ich je nach Lust, Laune und freier Zeit in der Zukunft verwirklichen will. Sollte jemand noch weitere interessante Themen im Bezug auf T-DSL via Satellit haben, kann er mir diese gerne zuschicken - vielleicht ja auch schon komplett ausformuliert. Ein Platz in der Danksagungsliste ist demjenigen sicher. ;-)

Internet-Zugang über T-DSL via Satellit unter Linux Version 0.4, 26. August 2002

Author: Wolfgang Wershofen, Systemberatung Wershofen

Auch wenn T-DSL via Satellit mittlerweile zum Regelangebot der Deutschen Telekom gehört, wird sich sicherlich im Laufe der Zeit noch einiges an diesem Dokument ändern. Schauen Sie öfters mal rein. ;-)

Die jeweils aktuellste Fassung dieses Dokumentes finden Sie unter  <http://www.wershofen.de/TDSLviaSAT-HowTo>