Linux ISDN HOWTO

Klaus Franken (i4l@klaus.franken.de) v1.5, 26. Mai 1999

Einrichtung eines Internet-Zugang-Rechners mit ISDN4Linux - Eine praxisorientierte Beschreibung mit Übungen.

Inhaltsverzeichnis

1 Einleitung **4**

1.1 Voraussetzungen . . . 4
1.2 Was soll erreicht werden? . . . 4
1.3 Was muß ich lesen, was soll ich lesen? . . . 5
1.4 Sprache . . . 5
1.5 keine Gewährleistung . . . 5
1.6 Feedback . . . 5
1.7 Copyright . . . 5

2 Grundlagen **6**

2.1 ISDN4Linux: Modem oder Netzwerk? . . . 6
2.2 Überblick über die Features . . . 6
2.3 Überblick über die fehlenden Features . . . 7
2.4 Überblick über die Tools . . . 7

3 Hardware-Modul laden **7**

3.1 isdnlog konfigurieren . . . 8
3.2 Plug&Play-Karten . . . 9
3.3 HiSax-Treiber laden . . . 11
3.3.1 Laden mit YaST . . . 11
3.3.2 Laden über `/etc/rc.config` . . . 11
3.3.3 Laden von Hand . . . 13
3.3.4 Troubleshooting . . . 13
3.4 Hardware testen . . . 14
3.5 Übung: Hardware ansprechen. . . . 14

4 Grundlagen ISDN, Parameter zur Verbindungskontrolle **15**

4.1 ISDN . . . 15
4.2 TK-Anlagen . . . 16
4.3 Was ist meine MSN? . . . 17
4.4 Probleme beim Verbindungsaufbau, die Cause Meldungen . . . 17

5 syncPPP Verbindung herstellen 18

5.1 Unterschiede analog - ISDN . . . 18

5.1.1 Analog: . . . 19

5.1.2 ISDN: . . . 19

5.2 Was ist eigentlich synchrones PPP . . . 19

5.3 Die Konfiguration . . . 19

5.3.1 Netzdevices anlegen und konfigurieren . . . 19

5.3.2 ipppd starten . . . 20

5.3.3 Authentifizierung beim ipppd . . . 22

5.3.4 Welche Daten muß ich über den Zugang kennen? . . . 24

5.3.5 PPP bei S.u.S.E. einrichten . . . 25

5.4 Probleme beim Verbindungsaufbau, Troubleshooting. . . . 26

5.5 Übung: syncPPP-Verbindung herstellen . . . 28

6 Probleme mit dynamischen IP-Nummern 28

6.1 Der RST-provoking mode . . . 30

6.2 Welche IP-Nummer setze ich denn eigentlich? . . . 30

7 Routing 31

7.1 Was ist Routing? . . . 31

7.2 Wie konfiguriert man das Routing? . . . 31

7.2.1 S.u.S.E. Methode . . . 32

7.2.2 Manuelle Methode . . . 32

7.2.3 Löschen von Routing-Einträgen . . . 32

7.3 Kontrollieren der Routingtabelle beim Verbindungsauf- und abbau (/etc/ppp/ip-up) . . . 33

7.3.1 Was macht das Script ip-up/ip-down? . . . 33

7.4 Übung: Kontrolliere die IP-Nummer und die Routing-Tabelle . . . 34

8 IP-Nummern Auflösung (DNS) 35

8.1 feste IP-Nummern Auflösung über /etc/hosts . . . 35

8.2 dynamische IP-Nummern Auflösung mit DNS . . . 36

8.3 Konfiguration der Namensauflösung . . . 36

8.3.1 Namensauflösung bei S.u.S.E. . . . 37

8.4 Probleme mit der Namensauflösung . . . 37

8.4.1 Checkliste . . . 37

9 Dial-On-Demand kontrollieren 38

9.1 Verbindungen überwachen . . . 38

9.2 Grund der Verbindung feststellen . . . 38

9.3 Verbindungen auswerten . . . 39
9.4 Dial-On-Demand an- und ausstellen . . . 39
9.5 Tips im S.u.S.E. System . . . 40
9.6 Wie erlaube ich normalen Benutzern Dial-In-Demand zu aktivieren? . . . 40

10 Konfiguration der Internet-Dienste 41
10.1 DNS-Cache . . . 41
10.2 Squid . . . 42
10.2.1 Starten von Squid . . . 42
10.2.2 Clients anpassen . . . 43
10.3 Fetchmail . . . 43
10.4 Sendmail . . . 44
10.5 News . . . 45
10.5.1 slrn installieren und konfigurieren . . . 45
10.5.2 Leafnode installieren und konfigurieren . . . 46
10.6 Firewall . . . 48
10.6.1 Was ist ein Paketfilter? . . . 48
10.6.2 Wie gibt man eine Firewall-Regel an? . . . 48
10.6.3 Was für Regeln brauche ich mindestens? . . . 49
10.6.4 Ein einfacher Firewall . . . 49
10.7 Masquerading . . . 50
10.8 Accounting . . . 50
10.9 Samba . . . 51

11 Installation 51
11.1 verwendete Programmversionen . . . 51
11.2 Unterschiede Kernel 2.0 und 2.2 . . . 51

12 Mailinglisten/News 51
12.1 Welche Mailinglisten gibt es? . . . 51
12.2 Wie frage ich auf der Mailingliste? . . . 52
12.3 Wie helfe ich auf der Mailingliste? . . . 52

13 Links 52
13.1 WWW und FTP . . . 52
13.2 lokale Dokumentationen . . . 53
13.3 Bücher . . . 53

14 Credits 54

15 News **54**

15.1 v1.2, 02. September 1998 54

15.2 v1.3, 18. Januar 1999 54

15.3 v1.4, 1. März 1999 54

15.4 v1.5, 26. Mai 1999 54

1 Einleitung

Das Tutorium wendet sich an ISDN-Einsteiger und solche mit ersten Erfahrungen, die sich jetzt auch für die weitere Konfiguration des Gesamt-Systems (z.B. Mailsystem, Firewalls, etc.) interessieren.

Das Tutorium wird praxisorientiert durchgeführt. Es werden nicht alle Grundlagen und Features im Detail besprochen, sondern der Teilnehmer hat nach dem Tutorium ein entsprechend konfigurierten Rechner bzw. die Grundlagen dazu.

Im Tutorium wird die Distribution *S.u.S.E. Linux 5.2* benutzt. Andere Distributionen (Debian, RedHat, ...) können selbstverständlich auch benutzt werden. Bei Bedarf werden die notwendigen Scripte installiert. Siehe dazu 11 (Installation).

In der S.u.S.E. Distribution sind zum einen die notwendigen Tools als auch Konfigurationsscripte enthalten, die eine abstraktere Konfiguration der ISDN-Verbindungen erlauben. Im Tutorium wird jeweils der *einfache* Weg über die Scripte und dann zur Referenz der manuelle Weg beschrieben.

1.1 Voraussetzungen

Der Teilnehmer sollte über Linux-Grundkenntnisse verfügen. Auf dem Rechner sollte die Basis-Installation schon erfolgreich durchgeführt sein.

Weiterhin sollte eine unterstützte ISDN-Karte eingebaut sein. Zu empfehlen ist z.B. eine AVM-Fritz classic oder eine ELSA QS1000. Siehe

http://www.suse.de/Support/sdb/isdn.html

für eine Liste der unterstützten Karten.

1.2 Was soll erreicht werden?

Folgende Aufgabe wird gelöst: Ein Linux-Rechner mit ISDN-Karte soll Internet-Zugangs-Rechner (IZG) werden. Der Rechner wählt sich bei einem Verbindungswunsch automatisch beim Internet-Service-Provider (ISP) ein und stellt die Netzverbindung transparent her. Benutzer an dieser Arbeitsstation haben nur vollständigen Zugriff auf das Internet und können z.B. WWW- und FTP-Dienste nutzen. Das Mailsystem wird so eingerichtet, daß beim Verbindungsaufbau automatisch die E-Mails ausgetauscht werden.

Ein eigenes Kapitel behandelt die Anbindung eines lokalen Netzwerkes mit vollständiger Internet-Nutzung (Masquerading, Mail, WWW, FTP-Nutzung) und der besonderen Probleme.

Da es sich um eine Wählleitung handelt, wird besonderes Augenmerk darauf gerichtet, daß zwar voller Internetzugang besteht, aber die Telefonkosten möglichst gering gehalten werden.

Um den roten Faden nicht zu verlieren, werden folgende Annahmen gemacht, die für die meisten Privatanwender (aber auch kleine Firmen, die nur einen *privaten* Internet-Zugang nutzen) zutreffen:

- ISDN Wählleitung ohne TK-Anlage (Euro-ISDN)
- Protokoll: syncPPP mit dynamischen IP-Nummern
- kein Proxy-Zwang
- Mails können über SMTP verschickt und POP3 abgeholt werden

Diese Voraussetzungen treffen inzwischen auf die meisten Privat-Zugänge z.B. T-Online oder Personal-Eunet und viele private Vereine zu.

Weiterhin wird auf sicherheitsrelevante Fragen, Probleme mit dynamischen IP-Nummern und den Anschluss eines lokalen Netzwerks an den IZR besprochen.

1.3 Was muß ich lesen, was soll ich lesen?

Der Text ist recht lang geworden, da an vielen Stellen auf besondere Probleme eingegangen wird und Tips zum Troobleshooting gegeben werden. Wer an diesen Stellen keine Probleme hat, braucht sich natürlich nicht damit aufzuhalten - es schadet aber auch nicht.

Ähnlich verhält es sich mit einigen Grundlagen, z.B. Routing oder das Konfigurieren spezieller Anwendungen, z.B. Mailaustausch. Der erfahrene Leser wird hier wenig Neues erfahren und kann diese Kapitel überlesen. Sie sind aber deswegen hier aufgenommen, weil das Verständnis hierfür unabdingbar ist und genau an diesen Punkten oft die meisten Probleme im Alltag auftauchen.

1.4 Sprache

Der Einfachheit und der besseren Lesbarkeit wegen werde ich Dich, den Leser, duzen.

1.5 keine Gewährleistung

Der Text ist nach bestem Wissen und Gewissen geschrieben. Der Autor übernimmt keine Gewährleistung, daß die hier vorgestellten Methoden richtig sind, funktionieren, sicher sind oder tatsächlich keine unnötigen Verbindungen aufgebaut werden.

Der Leser soll aber für ein einfaches System in die Lage versetzt werden, genau dies in den Griff zu bekommen :-)

1.6 Feedback

Erwünscht!

Per E-Mail an: *i4l@klaus.franken.de*

1.7 Copyright

Dieses Dokument ist urheberrechtlich geschützt. Das Copyright liegt bei Klaus Franken.

Das Dokument darf gemäß der GNU General Public License verbreitet werden. Insbesondere bedeutet dieses, daß der Text sowohl über elektronische wie auch physikalische Medien ohne die Zahlung von Lizenzgebühren verbreitet werden darf, solange dieser Copyright Hinweis nicht entfernt wird. Eine kommerzielle Verbreitung ist erlaubt und ausdrücklich erwünscht. Bei einer Publikation in Papierform ist das Deutsche Linux HOWTO Projekt hierüber zu informieren.

2 Grundlagen

Ich kenne mich mit Linux ein wenig aus. Ich kann mich vielleicht schon mit meinem Modem in's Internet einwählen.

Ich habe jetzt ISDN - und finde mich überhaupt nicht mehr zurecht. Warum eigentlich?

2.1 ISDN4Linux: Modem oder Netzwerk?

Vergiß alles, was Du über Modems weißt.

Bei ISDN ist alles anders:

1. Es klickt und pfeift nicht.
2. Es blinken keine Lämpchen.
3. Der Verbindungsaufbau geht so schnell, daß man innerhalb von Stunden ein Monatsgehalt loswerden kann.
4. Es macht mehr Spaß.

Die Konzepte unterscheiden sich:

1. Die Art der Hardwareanbindung
2. Die Art der Nutzung
3. Die Art der Kontrollmöglichkeiten
4. Die Art der Konfiguration

Bei ISDN4Linux wird die ISDN-Karte als Netzwerkkarte betrachtet, nur mit besonderen Eigenschaften.

2.2 Überblick über die Features

- schnelle ISDN-Verbindungen
- volle Integration als Netzwerk
- Client und Server
- syncPPP
- Modememulation (Befehlssatz)
- Dial-On-Demand (DoD)
- volle Übersicht und Kontrolle
- Callback
- Kanalbündelung

2.3 Überblick über die fehlenden Features

- serielle Anbindung (z.B. Fax)
- CAPI Schnittstelle (z.B. über Netz)
- einheitliche Umgebung...
- PmX-Karten

2.4 Überblick über die Tools

isdnlog

Der isdnlog *horcht* ständig im Hintergrund auf dem S0-Bus und protokolliert ein- und ausgehende Verbindungen zur späteren Auswertung (inkl. Gebühren) und zur Diagnose.

isdnctrl

Stellt wichtige I4l-spezifische Parameter (z.B. Telefonnummern) ein.

HiSax

Der Treiber (als Modul oder fest im Kernel) für fast alle passiven ISDN-Karten.

hisaxctrl

Kontrolliert den HiSax-Treiber

ipppd

Der PPP-Dämon für ISDN (syncPPP)

messages

In `/var/log/messages` (bzw. via Syslog) werden die ISDN-Aktionen protokolliert - wichtig zur Diagnose!

ttyI

Über die Terminal-Devices `/dev/ttyI0` bis `/dev/ttyI64` kann mit *normalen* Terminalprogrammen auf ISDN zugegriffen werden - Achtung: kein analoger Zugriff!

vbox

Der Anrufbeantworter für ISDN.

3 Hardware-Modul laden

Die Treiber für die Hardware werden durch Module bereitgestellt. Man könnte die notwendigen Treiber auch direkt in den Kernel laden, aber davon ist abzuraten.

Für das I4L-Subsystem ist das Modul `isdn` zuständig, das (je nach Compilierung) noch `slhc` benötigt.

Diese Module sind für den eigentlichen Hardwaretreiber Voraussetzung und müssen vorher geladen sein. Wenn die Module über das Tool `modprobe` lädt, braucht man sich darum aber nicht zu kümmern, da dadurch die Abhängigkeiten selbstständig geprüft werden.

Merke: Benutze nur `modprobe` zum Laden der Module.

Je nach verwendeter Hardware sind unterschiedliche Module notwendig. Für passive ISDN-Karten ist das Modul `HiSax` notwendig. Für aktive Karten werden herstellerspezifische Module benötigt.

3.1 isdnlog konfigurieren

Der isdnlog horcht ständig auf dem D-Kanal und liefert uns sowohl zur Diagnose, als auch zur später zur Auswertung wichtige Daten. Der isdnlog wird kurz nach dem Laden des HiSax-Treibers gestartet (bei aktiven Karten siehe unten).

Wir gehen später auf die Funktionen und den Start des isdnlog ein, hier nur kurz die wichtigsten Punkte zur Konfiguration:

- `/etc/isdn/isdn.conf` Es werden einige Daten über die Umgebung mitgeteilt, z.B. den Areacode (Vorwahl), den i4l nicht automatisch ermitteln kann.

  Passe in dem Bsp. zumindest den Areacode an:

```
# exapmle of /etc/isdn/isdn.conf
# copy this file to /etc/isdn/isdn.conf and edit
#
# More information: /usr/doc/packages/i4l/isdnlog/README


[GLOBAL]
COUNTRYPREFIX   = +
COUNTRYCODE     = 49
AREAPREFIX      = 0

# EDIT THIS LINE:
AREACODE        = 911

# Example:
#AREACODE        = 911 # Nuernberg

[VARIABLES]

[ISDNLOG]
LOGFILE = /var/log/isdn.log
ILABEL  = %b %e %T %ICall to tei %t from %N2 on %n2
OLABEL  = %b %e %T %Itei %t calling %N2 with %n2
REPFMTWWW  = "%X %D %17.17H %T %-17.17F %-20.20l SI: %S %9u %U %I %O"
REPFMTSHORT  = "%X%D %8.8H %T %-14.14F%U%I %O"
REPFMT  = "  %X %D %15.15H %T %-15.15F %7u %U %I %O"
```

- Optionen für isdnlog. isdnlog verträgt eine Menge Optionen, die man entweder als Kommandozeilenparameter, oder über eine Konfigdatei mitgeben kann.

  Bei S.u.S.E. wird die Datei `/etc/isdn/isdnlog.isdnctrl0.options` verwendet (0: erste Karte, 2: zweite Karte, 4: dritte Karte), und beim Start des isdnlog mit dem Parameter `-f` übergeben. Diese Datei ist kommentiert und enthält die wichtigsten Parameter.

  Mehr Infos gibt es in der README-Datei zu isdnlog, die in dem Quellpaket dabei ist, bei S.u.S.E. auch unter `/usr/doc/packages/i4l/isdnlog/README` eingepackt.

  Von Hand sollte isdnlog mit mindestens folgenden Option gestartet werden:

```
isdnlog -D -l1015 -x4087 -M -n -W80 /dev/isdnctrl0 &
```

- Telefonbuch (optional) isdnlog kann die ein- und ausgehenden Nummern automatisch einem Aliasnamen zuweisen, der statt der Telefonnummer angezeigt wird. Diese Daten stehen in: `/etc/isdn/callerid.conf`. Beispiel:

```
[NUMBER]
NUMBER = +4991152145922
ALIAS  = Eunet-N
ZONE   = 1
```

Darüber lassen sich auch weitere Aktionen definieren, z.B. Starten eines bestimmten Programmes beim Ring.

3.2 Plug&Play-Karten

PnP-Karten müssen im 2.0er Kernel noch manuell konfiguriert werden. Das ist etwas mühsam, muß aber zum Glück nur einmal gemacht werden.

Zum Konfigurieren unter Linux dient das Paket `isapnp`. Das zwei Programme bietet:

1. pnpdump: scannt den ISA-Bus nach Karten und erstellt eine Vorlage für die Konfigurationsdatei
2. isapnp: initialisiert die PnP-Karten entsprechend der Konfigurationsdatei.

Erst nachdem die Karte(n) hiermit konfiguriert wurde(n), kann durch Treiber auf die Hardware zugegriffen werden. PnP-Karten können also nur durch Modul - nicht durch Treiber im Kernel - benutzt werden.

Zuerst scannen wir nach PnP-Karten, aber Vorsicht: `pnpdump` **kann den Rechner zum Stillstand bringen**. Starte das Programm nicht unter X und möglichst nur im Single-User-Mode.

Die Ausgabe von `pnpdump` leiten wir gleich in die Konfigurationsdatei um:

```
pnpdump > /etc/isapnp.conf
```

Hier ein Beispiel für eine Elsa QS3000:

```
# This is free software, see the sources for details.
# This software has NO WARRANTY, use at your OWN RISK
#
# For details of this file format, see isapnp.conf(5)
#
# For latest information on isapnp and pnpdump see:
# http://www.roestock.demon.co.uk/isapnptools/
#
# Compiler flags: -DREALTIME -DNEEDSETSCHEDULER
#
# Trying port address 0203
# Board 1 has serial identifier e5 00 00 00 00 34 01 93 15

# (DEBUG)
(READPORT 0x0203)
(ISOLATE)
(IDENTIFY *)

# Card 1: (serial identifier e5 00 00 00 00 34 01 93 15)
# ELS0134 Serial No 0 [checksum e5]
# Version 1.0, Vendor version 0.0
# ANSI string -->ELSA QuickStep 3000<--
```

```
#
# Logical device id ELS0134
#
# Edit the entries below to uncomment out the configuration required.
# Note that only the first value of any range is given, this may be changed if r
equired
# Don't forget to uncomment the activate (ACT Y) when happy

(CONFIGURE ELS0134/0 (LD 0

# Multiple choice time, choose one only !

#     Start dependent functions: priority acceptable
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x0160
#             Maximum IO base address 0x0360
#             IO base alignment 16 bytes
#             Number of IO addresses required: 16
#(IO 0 (BASE 0x0160))
#       IRQ 3, 4, 5, 7, 10, 11, 12 or 15.
#             High true, edge sensitive interrupt (by default)
#(INT 0 (IRQ 3 (MODE +E)))

#     End dependent functions
#(ACT Y)
))
# End tag... Checksum 0x00 (OK)

# Returns all cards to the "Wait for Key" state
(WAITFORKEY)
```

Anhand der ausgegebenen Identifier kann man erkennen, welche Karten erkannt wurden (und ob es überhaupt PnP-Karten gibt)

Diese Datei wird editiert; die Kommentarzeichen müssen entfernt werden und ggf. passende Werte eingesetzt werden. In den Kommentaren werden gültige Werte angegeben.

```
(IO 0 (BASE 0x0160))
(INT 0 (IRQ 3 (MODE +E)))
(ACT Y)
```

Man beachte: **Auch `(ACT Y)` muß gesetzt werden**. Ansonsten passiert gar nichts.

Diese Konfiguration kann nun auf die PnP-Karte heruntergeladen werden:

```
isapnp /etc/isapnp.conf
Board 1 has Identity e5 00 00 00 00 34 01 93 15:  ELS0134 Serial No 0 [checksum e5]
```

Die Ausgabe ist leider nicht sehr aufschlußreich, aber man sollte zumindest den Identifier der Karte erkennen.

Bei S.u.S.E. wird das `isapnp` Kommando automatisch in den Init-Scripten ausgeführt. Ansonsten muß man selbst für diesen Aufruf sorgen.

3.3 HiSax-Treiber laden

Dem HiSax-Treiber wird durch Parameter beim Laden mitgeteilt, nach welcher Karte (oder auch Karten) an welchen Adressen zu suchen ist.

3.3.1 Laden mit YaST

Bei S.u.S.E. kann die Konfiguration der ISDN-Hardware mittels YaST in der Maske `Administration des Systems/ Hardware in System integrieren/ ISDN-Hardware konfigurieren` vorgenommen werden. Neben der Auswahl der Karte und setzen der Parameter kann hier auch sofort das Modul geladen werden durch `Starten`. Bei Problemen kann man sofort andere Werte probieren. Bei Erfolg werden die Parameter mittels `Speichern` in `rc.config` abgelegt, so daß die Module beim nächsten Systemstart wieder geladen werden.

Die Syntax ist in

/usr/src/linux/Documentation/isdn/README.HiSaX

beschrieben.

3.3.2 Laden über `/etc/rc.config`

Die ISDN-Hardware kann direkt in der `/etc/rc.config` eingetragen und/oder kontrolliert werden. Die Variablen sind kommentiert. Hier ein Beispiel für eine Elsa QS-3000:

```
#
# start i4l? ("yes" or "no")
#   see: /usr/doc/packages/i4l/README.SuSE
#
I4L_START="yes"

#
# driver-id for HiSax-driver
#   set to "HiSax"
#   or whatever you defined when loading driver within kernel
#   set to "" if you don't have a hisax-card
#
I4L_TELES_ID="hisax1"

#
# D-channel protocol 1=1TR6, 2=EDSS1(Euro-ISDN) for HiSax
#
I4L_PROTOCOL="2"

# Type   ISDN-card               Required parameters
# ----   --------------------    ------------------------------------------
#    1   Teles 16.0              irq, mem, io
#    2   Teles  8.0              irq, mem
#    3   Teles 16.3 (non PnP)    irq, io
#    4   Creatix/Teles PnP       irq, io0 (ISAC), io1 (HSCX)
#    5   AVM A1 (Fritz)          irq, io
#    6   ELSA PCC/PCF cards      io or nothing for autodetect (the iobase is
#                                only required, if you have more than one ELSA
```

```
#                                card in your PC)
#    7   ELSA Quickstep 1000     irq, io  (from isapnp setup)
#    8   Teles 16.3 PCMCIA       irq, io
#    9   ITK ix1-micro Rev.2     irq, io
# since: HiSax 2.5:
#   10   ELSA PCMCIA             irq, io  (set with card manager)
#   11   Eicon.Diehl Diva ISA PnP irq, io
#   11   Eicon.Diehl Diva PCI    no parameter
#   12   ASUS COM ISDNLink       irq, io  (from isapnp setup)
#   13   HFC-2BS0 based cards    irq, io
#   15   Sedlbauer Speed Card    irq, io
#        (= Teledat 100)
#   16   USR Sportster internal  irq, io
#   17   MIC card                irq, io
#   18   ELSA Quickstep 1000PCI  no parameter
#
I4L_TELES_TYPE="7"

#
# IRQ of Teles Card
#   eg. 12 or 15 when loading as module
#   set to "" when driver is loaded within kernel
#
I4L_TELES_IRQ="3"

#
# Portaddress of Teles card (e.g. 0xd80, "0" for S0/8)
#
I4L_TELES_PORT="0x0160"
```

Der String `TELES` hat hier nur historische Gründe.

Mit diesen Angaben wird die Parameterzeile für den HiSax selbständig generiert. Zusätzlich kann man auch die Parameterzeile komplett selbst vorgeben, was z.B. bei neuen Karten notwendig ist, oder wenn man mehrere Karten anbinden will (s.u.).

Beispiel für eine AVM-Fritz und eine ELSA PCF Karte:

```
I4L_TELES_MODUL_OPTIONS="type=5,6 protocol=2,2 io=0x340 irq=10 id=Fritz%Elsa"
```

Zum Laden der Module benutzt man dann Init-Script:

```
glen:/root # /sbin/init.d/i4l_hardware start
Loading ISDN drivers ...
Loading HiSax driver ...
/sbin/insmod /lib/modules/2.0.33/misc/hisax.o id=hisax1 type=7 protocol=2 irq=3 io=0x0160
Verbose-level set to 3.
Starting isdnlog with /etc/isdn/isdnlog.isdnctrl0.options for isdnctrl0...
```

Man beachte, daß hiermit automatisch der `isdnlog` gestartet wird.

Zum Entladen benutze man dasselbe Script:

```
glen:/root # /sbin/init.d/i4l_hardware stop
Unloading ISDN drivers ...
```

3.3.3 Laden von Hand

Die Syntax ist in

/usr/src/linux/Documentation/isdn/README.HiSax

beschrieben.

Für eine ELSA-QS3000 gebe man z.B. ein:

```
modprobe -v hisax id=hisax1 type=7 protocol=2 irq=3 io=0x0160
```

Weiterhin sollten nach dem erfolgreichen Laden folgende Kommandos ausgeführt werden:

```
/sbin/hisaxctrl hisax1 1 4
/sbin/isdnctrl verbose 3
/sbin/isdnlog /dev/isdnctrl0
```

Erklärt werden diese Kommandos in den entsprechenden man-pages und mitgelieferter Doku, mit dem S.u.S.E. Scripts ist es halt einfacher ;-)

3.3.4 Troubleshooting

Während des Ladens des Hisax-Moduls bekommt man im Fehlerfall auf der Konsole keine aussagekräftigen Meldungen, sondern meist nur `Device or resource busy`. Die echten Fehlermeldungen werden via Syslog (zumeist) in */var/log/messages* gespeichert.

Beispiel für einen Mißerfolg beim Laden einer AVM-Fritz:

```
Feb  6 22:45:05 glen kernel: HiSax: Driver for Siemens chip set ISDN cards
Feb  6 22:45:05 glen kernel: HiSax: Version 2.1
Feb  6 22:45:05 glen kernel: HiSax: Revisions 1.15/1.10/1.10/1.30/1.8
Feb  6 22:45:05 glen kernel: HiSax: Total 1 card defined
Feb  6 22:45:05 glen kernel: HiSax: Card 1 Protocol EDSS1 Id=HiSax (0)
Feb  6 22:45:05 glen kernel: HiSax: AVM driver Rev. 1.6
Feb  6 22:45:05 glen kernel: AVM A1: Byte at 1b00 is ff
Feb  6 22:45:05 glen kernel: AVM A1: Byte at 1b03 is ff
Feb  6 22:45:05 glen kernel: AVM A1: Byte at 1b02 is ff
Feb  6 22:45:05 glen kernel: AVM A1: Byte at 1b00 is ff
Feb  6 22:45:05 glen kernel: HiSax: AVM A1 config irq:12 cfg:1b00
Feb  6 22:45:05 glen kernel: HiSax: isac:1700/1300
Feb  6 22:45:05 glen kernel: HiSax: hscx A:700/300  hscx B:f00/b00
Feb  6 22:45:05 glen kernel: AVM A1: HSCX version A: ???  B: ???
Feb  6 22:45:05 glen kernel: AVM A1: ISAC 2085 V2.3
Feb  6 22:45:05 glen kernel: AVM A1: wrong HSCX versions check IO address
Feb  6 22:45:05 glen kernel: HiSax: Card AVM A1 not installed !
```

Hier wurde an der angegebenen Portadresse keine Fritz-Karte gefunden. (Es war auch keine vorhanden ;-). Anhand dieser Meldungen sollte man leicht erkennen können, was die genaue Ursache ist. Weitere häufige Fehler sind:

1. `could not get interrupt`: mit dem angegebenen IRQ kann nicht gearbeitet werden. Probiere einfach einen anderen. Nicht belegte IRQ kann man durch `cat /proc/interrupts` ermitteln.
2. Portadresse wird nicht erkannt, obwohl alles richtig scheint: Es ist eine PnP-Karte, isapnp wurde vergessen. Siehe 3.2 (Plug&Play-Karten).
3. Portadresse wird nicht erkannt, obwohl alles richtig scheint: es ist eine Teleskarte, man kann also nicht wissen, um welchen Typ es ist wirklich handelt. Abhilfe: neuesten HiSax besorgen und alles ausprobieren. Siehe 11 (Installation).

Bei hartnäckigem Mißerfolg, wende Dich an einen guten Bekannten oder an die Mailingliste. Unbedingt den Ausschnitt aus `/var/log/messages` angeben!

3.4 Hardware testen

Der beste und einfachste Test, ist sich selber anzurufen.

Es spielt hierbei keine Rolle, ob man ISDN-Daten- oder Voice-Call, ob von einem internen oder externen Analog- oder ISDN-Telefon anruft. Es wird auch keine Verbindung zu Stande kommen. Wichtig ist nur, daß man unter */var/log/messages* eine Meldung über den Anruf erkennt.

Beispiel für einen Analog-Call auf der MSN 123459:

```
Apr  6 22:15:20 glen kernel: isdn_net: call from 911123458,1,0 -> 123459
Apr  6 22:15:20 glen kernel: isdn_net: Service-Indicator not 7, ignored
```

Bei diesem Beispiel handelt es sich um ein Voice-Call (Service-Indicator: 0) von einem Anschluß mit Rufnummerübermittlung von der MSN 123458 aus dem Ortsnetz 0911 an die eigene MSN 123459. (Nein, das ist nicht meine echte Nummer ;-)

Wichtig ist vor allem hier die Angabe der Zielrufnummer hinter dem Pfeil, hier 123459. Man sollte hier alle eigenen Nummern durchprobieren. So wie es dort angegeben ist, ist auch später die eigene MSN zu setzen.

3.5 Übung: Hardware ansprechen.

Ziel: Die ISDN-Karte soll angesprochen und geprüft werden.

1. Welche Hardware / Umgebung hab ich? Notiere Dir:
   (a) Welche Karte hab ich (Hersteller, Typ, etc.)?
   (b) Wie ist die Karte gejumpert (Port)?
   (c) Mit welchen Werten kann die Karte unter anderen System angesprochen werden?
   (d) Welches Protokoll wird auf dem S0-Bus benutzt (1TR6, DSS1)?
   (e) **Wo** ist die ISDN-Karte angeschlossen (NTBA, TK-Anlage)?
   (f) Welche MSN's kann ich auf diesem S0-Bus benutzen?

   Schlimmstenfalls mußt Du Deinen Rechner aufschrauben, das falsche Betriebssystem booten und/oder den Administrator nerven.

2. Betrachte messages Nur in `/var/log/messages` steht die Wahrheit, sie ist für die gesamte Konfigurationsarbeit (und später) zu verfolgen.

   Öffne (mindestens) zwei Konsolen (virtuelle Linux-Konsole oder unter X zwei `xterm`).

   Auf einer Konsole starte entweder:

   - `tail -f /var/log/messages`
   - `less /var/log/messages`, im Programm dann `F` (follow) drücken, um immer die neuesten Zeilen zu bekommen. (Diesen Modus beendet man durch `Ctrl-C`, beenden von less mit `q`.

3. PnP Karte? Falls es eine Plug&Pray-Karte ist, konfiguriere sie, wenn Du es nicht weißt, starte `pnpdump`. Siehe 3.2 (Plug&Play-Karten).

4. Modul laden Lade das entsprechende Modul nach Deiner bevorzugten Methode (also YaST ;-).

   Stelle sicher, daß die Einstellungen notiert sind und beim Systemstart automatisch das Modul wieder geladen wird.

   Prüfe, ob das Modul geladen ist mit `lsmod`.

   Prüfe, ob der `isdnlog` läuft mit `/ps ax|grep isdnlog`

   Prüfe, ob `/var/log/messages` *normal* aussieht.

   Siehe 3.3 (HiSax-Treiber laden)

5. ISDN testen Rufe Dich selbst an und notiere alle MSN unter denen Du angerufen werden kannst.

   Siehe 3.4 (Hardware testen)

4 Grundlagen ISDN, Parameter zur Verbindungskontrolle

Ein Rundumschlag über die wichtigsten Begriffe und Konzepte, die man wissen muß, um ISDN unter Linux richtig zu nutzen.

4.1 ISDN

`ISDN` steht für **Integrated Services Digital Network**.

Fangen wir von hinten an: Es handelt sich um ein **Netzwerk**. Über die beiden Kupferdrähte wird also z.B. nicht nur eine Point-To-Point Verbindung aufgebaut, sondern es können mehrere Verbindungen gleichzeitig bestehen,

Die Daten werden alle **digital** ausgetauscht. Analogdienste wie z.B: Fax sind hierüber daher potentiell schwieriger zu handeln. Normalerweise werden Analogdienste über Spezialgeräte wie a/b-Wandler oder TK-Anlagen gefahren.

Integrated Services deutet an, daß verschiedene Dienste über dieses Netzwerk behandelt werden können. Typische Services sind **Analog-Sprache** (SI=0) oder **ISDN-Daten** (SI=7) was uns hier interessiert.

Der Endpunkt der Telekom-Leitung ist der **NTBA** (kurz auch **NT**), der **Network Terminator Basis-Anschluß**. Das ist der kleine Kasten, bis zu dem sich der Netzanbieter zuständig fühlt.

An einem NTBA können (normalerweise) 2 Kabel herausgeführt werden, diese bilden gemeinsam ein Bus-System, den sogenannten **S0-Bus**

An den S0-Bus können 8 **Endgeräte** angeschlossen werden. Typische Endgeräte sind ISDN-Telefone, TK-Anlagen G4-Fax-Geräte, ISDN-Terminaladapter und ISDN-Karten.

Der S0-Bus bietet 3 Kanäle: ein Steuerkanal, genannt **D-Kanal**. Weiterhin stehen zwei Datenkanäle, genannt **Nutzkanal** oder **B-Kanal** mit jeweils 64 kbit/s zur Verfügung, die jeweils zu unterschiedliche Partner und mit unterschiedlichen Diensten genutzt werden können.

Die Bezeichnung der Kanäle (laut **Technik der Netze** von Gerd Siegmund, 3. Aufl. S. 316): *Die Bezeichnungen für die Kanaltypen (B- und D-Kanal) sind ursprünglich keine Abkürzungen mit tieferen Bedeutungen. In der Anfangszeit der ISDN-Standardisierung wurden Kanaltypen mit verschiedenen Aufgaben und Eigenschaften betrachtet, die fortlaufend als A-, B-, C-, D- und E-Kanal bezeichnet wurden (A- Analog, B- 64 kbit*s digital, C- Komprimiert, D- Signalisierungskanal)./

Auf dem D-Kanal können verschiedene Protokolle gefahren werden. Üblich sind **1TR6** (altes nationales ISDN), **DSS1** (**Euro-ISDN**, der Quasi-Standard in 24 Ländern) und **N1** in den USA. Der D-Kanal dient u.a. zur Übermittlung des Wunsches eines Verbindungsauf- und abbaus, der Übermittlung der Telefonnummern und der Gebühren. Bei einem falsch eingestellten Protokoll klappt also sehr wenig...

Die Art und Weise, wie die Telefonnummer gemeldet und genannt wird, hängt vom D-Kanal-Protokoll ab:

- 1TR1: **EAZ** (Endgeräte-Auswahl-Ziffer). Es handelt sich also nur um eine Ziffer, die Rufnummer des Basisanschlusses wird nicht betrachtet.
- DSS1: **MSN** (Multiple-Subscribe-Number). Hier ist eine komplette Rufnummer gemeint, also alles hinter der Vorwahl.

Die Bezeichnungen EAZ und MSN sind bei I4L ansonsten synonym zu benutzen (wenn das richtige Protokoll angegeben wurde). Bei einem einkommenden Call wird (hoffentlich) die Zielrufnummer übertragen, genannt **CPN** (called party number). Ist sie nicht bekannt, setzt sie I4L auf `0`.

Bekanntlich können für einen Anschluß mehrere Telefonnummern vergeben werden. Diese signalisiert die Vermittlungsstelle (kurz **VSt**) auf dem D-Kanal (CPN) zusammen mit dem **Service-Indikator** (**SI**). Mehr passiert bei einem ankommenden Call erst mal nicht! Es ist danach Aufgabe der Endgeräte sich entsprechend zu verhalten: ignorieren, abweisen, oder den Call annehmen.

Da der SI zusammen mit der Nummer auf dem D-Kanal übermittelt wird, kann dieselbe Telefonnummer mehrfach genutzt werden. Beispiel: das Telefon reagiert nur auf SI=0, der PC reagiert nur auf SI=7.

Bei einem ausgehenden Call muß das Endgeräte die MSN angeben; diese wird dann auch dem Partner übermittelt. Wird keine MSN gesetzt (was I4L nicht zuläßt), setzt die VSt die Nummer ein, wird eine falsche MSN gesetzt, bekommt man keine Verbindung (Erfahrungswerte).

Mehr zu ISDN-Grundlagen findet sich auf

http://www.dtag.de/angebot/isdn/lexikon/right.htm

4.2 TK-Anlagen

Worum geht es: Wer die Wahl hat zwischen einem direkten Anschluß am NTBA und einem internen S0-Bus an einer TK-Anlage, sollte sich für den direkten Anschluß entscheiden! Der Betrieb über TK-Anlagen birgt immer gewisse Überraschungen.

Worum geht es nicht: Wenn man eine TK-Anlage am selben NTBA (S0 Bus) wie die ISDN-Karte angeschlossen hat, gibt es keine Probleme. Die TK-Anlage ist hier nur ein normales ISDN-Endgerät, was dieses *hinten* macht (Anschluß von Analog-Geräten) spielt hier keine Rolle.

Das Verhalten der TK-Anlage hängt unter anderem vom Typ, von der installierten Software und vor allem von deren Konfiguration (und damit vom entsprechenden Service-Techniker) ab.

Bei älteren Anlagen wird oft entgegen allen Aussagen 1TR6 anstatt DSS1 gefahren. Die Verbindungstypen können abhängig vom Service-Indikator konfiguriert werden, wobei oft nur Voice-Calls konfiguriert sind. Weiterhin besteht die Schwierigkeit herauszufinden, welche MSN/EAZ man zu benutzen hat.

Bevor man sich auf andere verläßt, sollte man den Praxistest *Selbstanruf* machen, siehe: 3.4 (Hardware testen)

Ein wesentlicher Unterschied ist der, daß man nicht mit allen anderen lokalen Teilnehmern an einem Bus angeschlossen ist, sondern die TK-Anlage für jeden einzelnen Anschluß einen eigenen S0-Bus nach außen führt, an den meist nur ein Endgerät angeschlossen wird. Dieser Anschluß bekommt eine eigene Durchwahl zugewiesen, oft 2-stellig.

Die beste Veranschaulichung ist die, daß man sich seine TK-Anlage als eine eigene Vst vorstellt.

Beispiel: In Ortsnetz 321 ist eine TK-Anlage mit der Rufnummer 654 an einem Primärmultiplex-Anlagenanschluß installiert (es gibt also mehr als 2 Amtsleitungen, alternativ könnte dies auch ein Bündelanschluß sein - spielt aus dieser Sicht keine Rolle). Es sind 20 interne Leitungen vorhanden, wobei die ersten 10 für Telefone und die zweiten 10 für ISDN-Karten vorgesehen sind. Die Durchwahlnummern seien 10-19 für die Telefon und 20-29 für die ISDN-Karten. Die S0-Busse für die ISDN-Karten seien auf DSS1 konfiguriert.

Dann ist als MSN jeweils 20 bis 29 zu benutzen, denn das sind die MSN's im Ortsnetz *Firma* (=321654). Weiterhin ist zu beachten, daß man zusätzlich eine `0` wählen muß, um aus dem Ortsnetz *Firma* erstmal herauszukommen. Um z.B. die Nummer 987 im Ortsnetz 321 anzurufen, muß man `0987` wählen, wobei der Gegenstelle als Rufnummer `65420` angezeigt wird. Will man in Berlin anrufen, wählt man selbst die `0030....` an und dort wird `32165420` übermittelt.

Will man selber User-Authentication beim Dial-In über die Telefonnummer machen, gibt es nur eine sinnvolle herangehensweise: anrufen lassen. Die in `/var/log/messages` angezeigte Nummer dann mittels Cut&Paste in die entsprechende Konfigurationsdatei übernehmen.

4.3 Was ist meine MSN?

Wie oben erwähnt, muß man bei I4L immer die MSN setzen, um wählen zu können, die Angabe der MSN ist wichtig, da ansonsten meist nichts funktioniert.

Die erste Frage ist dabei immer, ob man direkt am NTBA oder an einer TK-Anlage angeschlossen ist.

- Anschluß an NTBA: Man kann sich eine der 3 oder mehr zugewiesenen MSN's aussuchen. Diese MSN wird der Gegenstelle übermittelt. Wird die MSN zur Überprüfung des Partners benutzt (z.B. bei rawip), muß man sich mit der Gegenstelle natürlich fest auf eine einigen. Ansonsten hat man die freie Wahl, man kann durchaus seine normale Voice-Nummer benutzen.
- Anschluß an TK-Anlage: Man ist auf die Konfiguration der TK-Anlage angewiesen. Die einfachste Methode ist der Selbsttest, siehe 3.4 (Hardware testen)

4.4 Probleme beim Verbindungsaufbau, die Cause Meldungen

Das Protokoll auf dem D-Kanal erlaubt es Meldungen zu verschicken, die über den Grund bei einem Verbindungsabbruch und nicht erfolgreichem Verbindungsaufbau informieren.

Die Meldungen werden in `/var/log/messages` vom i4l-Subsystem als sogenannte `cause`-Meldungen weitergegeben.

Die Art der Meldung hängt vom verwendeten Protokoll ab (1TR6 oder DSS1), bei DSS1 wird ein `E` (für Euro-ISDN) vorangestellt, dahinter folgen vier (Hex-)Ziffern. Die ersten beiden geben Auskunft darüber, wo

diese Meldung generiert wurde (bei welcher VSt); die letzten beiden Ziffern geben den eigentlichen Grund an.

Der isdnlog übersetzt uns freundlicherweise die Meldungen in Klartext; wenn der nicht läuft (z.B. bei aktiven ISDN-Karten), kann man die Meldungen mittels `man isdn_cause` auflösen.

Nicht alle Meldungen müssen *dramatisch* sein und müssen auf einen Fehler hinweisen.

Beispiele:

```
kernel: isdn: hisax1,ch0 cause: E0010
kernel: ippp0: remote hangup
```

Ursache: der Gegner hat *normal* aufgelegt, vermutlich wegen Timeout.

```
kernel: isdn: hisax1,ch0 cause: E0511
isdnlog: Mar 19 20:00:32 tei 70 calling Leibnitz with
    Kfr User busy (Private network serving remote user)
```

Ursache: die VSt beim Gegner teilt uns mit, daß dort der Anschluß besetzt ist.

```
kernel: isdn: hisax1,ch0 cause: E0022
isdnlog: Mar 19 21:37:16 tei 70 calling Klein with +49 911/
    333, N|rnberg  No circuit/channel available (User)
```

Ursache: alle Kanäle sind belegt.

```
kernel: isdn0: dialing 1 1111111111...
isdnlog: Apr 13 15:05:18 * tei 84 calling +49 911/111111111,
    N|rnberg with Kfr  RING (Data)
kernel: isdn: hisax1,ch0 cause: E0201
isdnlog: Apr 13 15:05:19 * tei 127 calling ? with ?  Unallocated
    (unassigned) number (Public network serving local user)
```

Ursache: die Zielrufnummer ist nicht zugewiesen.

5 syncPPP Verbindung herstellen

Point-to-Point Protocol (PPP) ist u.a. in den RFCs 1332, 1334, 1570, 1661, 1662, 1994 definiert.

Es wurde ursprünglich entwickelt, um Daten über serielle Leitungen zu verschicken. Es bietet grundsätzlich auch weitere Netzwerkprotokolle (Apple, IPX, ...) unter Linux ist aber nur IP vorgesehen. PPP bietet verschiedene Features wie z.B. den Austausch der IP-Nummern, Authentication, Compressions etc.

Aus diesem Grund findet zunächst durch das Link Control Protocol (LCP) ein Handshake statt, durch den die Verbindung initialisiert wird (oder auch nicht). In der Praxis ergeben sich genau hierdurch die Probleme gemäß der Richtlinie *das schöne an Standards ist, daß sich jeder seinen eigenen aussuchen kann.*

5.1 Unterschiede analog - ISDN

Wer analoges PPP gewöhnt ist, muß hier ein umdenken. Bei ISDN dreht sich alles um. Die Netzverbindung besteht logisch immer, gewählt wird aber nur bei Bedarf.

5.1.1 Analog:

- manuelles Starten eines Scriptes oder über diald
- wählen, z.B. mit 'chat'
- pppd fährt hoch und macht Handshake mit Partner
- ifconfig und route Aufrufe durch pppd
- Optionsfile: `/etc/ppp/options`
- liest automatisch `/etc/ppp/options.DEVICE` (DEVICE ist das aktuell verwendete serielle Device).

5.1.2 ISDN:

- Netz wird konfiguriert, diverse isdnctrl und ein ifconfig Aufruf
- Setzen der Route
- starten ipppd
- Verbindungswunsch -> i4l wählt selbständig Nummern (isdnctrl)
- ipppd wird aktiviert (er läuft die ganze Zeit!)
- ipppd macht Handshake
- Bei dynamischen IP-Nummern legt der ipppd das Device neu an und startet `/etc/ppp/ip-up`
- Beim (automatischen Abbau) macht der ipppd ein Reconnect auf das Device bei analog beendet er sich.
- Beim Abbau startet der ipppd `/etc/ppp/ip-down`
- Optionsfile: `/etc/ppp/ioptions`
- Liest kein weiteres Optionfile automatisch ein.

5.2 Was ist eigentlich synchrones PPP

Der Unterschied zwischen synchronem und asynchronem PPP ist das **Framing** also das Einpacken der Rohdaten für die jeweilige Verbindungsart. SyncPPP packt in HDLC ein. Auf einer Modemstrecke bzw. einer seriellen Schnittstelle kann man aber nur characterweise verschicken. Entsprechend packt asyncPPP seine Päckchen anders ein. Es gibt ein ausgewiesenes Byte welches den Paketanfang bzw. das Ende markiert. Diese Byte muss, sofern es im Datenstrom auftaucht natürlich anders dargestellt werden. Dafür gibt es ein weiteres reserviertes Byte, das Escape-Byte.

5.3 Die Konfiguration

5.3.1 Netzdevices anlegen und konfigurieren

Beispiel:

```
NETDEV='ippp0'
# neues Device
isdnctrl addif $NETDEV

# setzte MSN/EAZ
isdnctrl eaz $NETDEV 456

# def. Nummer(n) zum rauswaehlen
isdnctrl addphone $NETDEV out 09011

# erlaube Nummern, die anrufen duerfen
#isdnctrl addphone $NETDEV in

# duerfen alle anrufen? Nein: setze secure=on
isdnctrl secure $NETDEV on

# Layer-2 Protokoll: (x75i, x75ui, x75bui, hdlc)
isdnctrl l2_prot $NETDEV hdlc

# Layer-3 (nur trans)
isdnctrl l3_prot $NETDEV trans

# Ecapsulation: (rawip, cisco_h, syncppp)
isdnctrl encap $NETDEV syncppp

# Idletime
isdnctrl huptimeout $NETDEV 60

# maximale Waehlversuche
isdnctrl dialmax $NETDEV 5

# nur einen bestimmten Kanal benutzen
#isdnctrl bind $NETDEV

# PPP an Netzdevice binden
isdnctrl pppbind $NETDEV 0

# Netzdevice konfigurieren
ifconfig $NETDEV 1.1.1.1 pointopoint 193.102.150.13 up

# OPEN-Meldung ausgeben:
isdnctrl verbose 3
```

Gelöscht wird das Interface durch die Befehle:

```
ifconfig $NETDEV down
isdnctrl delif $NETDEV
```

5.3.2 ipppd starten

Vor dem Start des ipppd müssen drei Voraussetzungen erfüllt sein:

1. ISDN-Hardwareunterstützung

2. syncPPP Untersützung im Kernel
3. Das zu verwendende Device muß angelegt sein (`isdnctrl addif`)

Die syncPPP Unterstützung des Kernels prüft der ipppd leider über eine etwas unglücklich Methode ab: Es muß ein Device `ippp0` vorhanden sein. Außerdem kann man das Device nicht beliebig benennen, es muß mit `ippp` beginnen. Merke: **Benutze immer als Devicename** `ippp0`

Der ipppd kann über 2,5 Arten Optionen annehmen:

1. Kommandozeilenparameter
2. Das Optionsfile `/etc/ppp/ioptions`

Die 2,5te Methode ist die Angabe eines Optionsfile als Kommandozeilenparameter (`-file`).

In Anlehnung an den pppd empfehle ich folgende Struktur:

- Globale Optionen (für alle ipppd's) in `/etc/ppp/ioptions`
- Devicespezifische Optionen (z.B. für ippp0) in `/etc/ppp/options.ippp0`
- Start des ipppd mit

```
ipppd pidfile /var/run/ipppd.ippp0.pid file /etc/ppp/options.ippp0 &
```

Folgendes sollte man noch über den ipppd wissen:

- Es werden z.T. andere Optionen als beim pppd benutzt, zu den Unterschieden siehe `man ipppd`
- Die Optionen und Paßwörter werden nur beim Start neu eingelesen. Beim Testen sollte man also immer nachschauen, ob noch ipppd'd laufen und neustarten.
- Es können verschiedene ipppd auf ein Device reagieren, daher `pppbind`.
- Die Datei `/etc/ppp/ioptions` muß existieren.

Folgende Optionen haben sich für die verschiedensten ISP's als stabil erwiesen:

```
# /etc/ppp/options.ippp0
#
# for isdn4linux/syncPPP and dynamic IP-numbers
#
#
# Klaus Franken, kfr@suse.de
# Version: 27.08.97 (5.1)
# Umstellung auf CHAP: 26.05.99 kfr@klaus.franken.de
#
# This file is copy by YaST from /etc/ppp/ioptions.YaST
#   to options.<device>

# The device(s)
# for more than one device try:
# /dev/ippp0 /dev/ippp1 ...
/dev/ippp0
```

```
# The IP addresses: <local>:<remote>
# just "0.0.0.0:" or nothing for dynamic IP
#0.0.0.0:

# my user name (to force PAP instead of CHAP)
# user suse

# my system name
name suse

# accept IP addresses from peer
# use with dynamic IP
ipcp-accept-local
ipcp-accept-remote
noipdefault

# try to get IP address from interface
# option specific to ipppd (as opposed to pppd)
# use only with static IP
#useifip

# disable all header-compression
-vj
-vjccomp
-ac
-pc
-bsdcomp

# sometimes you need this:
#noccp

# max receive unit
mru 1524
# max transmit unit
mtu 1500

# If this machine is a server, force authentication by uncommenting one
# of the following. However, if this machine is a client, doing this will
# prevent a succesful connection! (message "peer refused to authenticate").
# So, only uncomment on a server.
# "+pap" / "+chap" NUR AKTIVIEREN, WENN DIES EIN SERVER IST!!!
#+pap
#+chap

# if you have problems with handshaking (no response for first
# lcp-package) try to decrease the retry-cycle. Default is 3 sec,
# try for example 2 sec:
# lcp-restart 2
```

5.3.3 Authentifizierung beim ipppd

Der ipppd verhält sich bei der Benutzerauthentifizierung exakt genauso wie der pppd. Daher nur ein kurzer Abriß.

Es stehen zwei Methoden zur Verfügung: PAP und CHAP. Hier wird es so konfiguriert, daß automatisch beides funktioniert. Siehe auch 5.3.3 (PAP oder CHAP?).

Die Benutzerdaten werden an zwei Stellen konfiguriert; als erstes wird dem ipppd durch das Schlüsselwort `name` mitgeteilt, welche User-Id dem gegnerischen PPP-Daemon angeboten werden soll.

Als nächstes wird das Passwort selbst (im Klartext) in der Datei `/etc/ppp/pap-secrets` abgelegt.

Für jeden verwendeten User wird eine Zeile eingetragen, z.B. sei der Username `suse` und Passwort `linux`:

```
# client        server  pw              iplist
"suse"          *       "linux"
```

Diese Datei darf nur für root lesbar sein! Also

```
chmod 600 /etc/ppp/pap-secrets
```

Dies ist die einzige Stelle, wo das Passwort definiert ist. In der `/etc/ppp/pap-secrets` können mehrere User/Passwörter definiert sein, über die Option `user` bzw. `name` wird jeweils die richtige Zeile extrahiert, um das Passwort zu ermitteln.

Als nächstes wird die `chap-secrets` auf die `pap-secrets` gelinkt:

```
cd /etc/ppp
ln -s pap-secrets chap-secrets
```

Der Benutzername muß nicht im System bekannt sein, zumindest besteht zwischen dem PAP- (oder CHAP-) Benutzernamen und dem Systembenutzer kein Zusammenhang.

Nachdem der ipppd gestartet ist und ev. eine Route darüber definiert ist, wird bei Bedarf automatisch ein Wählvorgang ausgelöst. Manuell kann man dies auslösen durch `isdnctrl dial ippp0`. Meldungen werden über syslog nach `/var/log/messages` geschrieben.

Sonderzeichen im Usernamen oder Passwort Normalerweise brauchen Username und Passwort nicht in Anführungszeichen eingeschlossen werden.

Sind aber Sonderzeichen wie z.B. # darin enthalten, muß man die Anführungszeichen setzen.

Enthält der Username oder das Passwort allerdings das Anführungszeichen selbst, dann darf man widerum keine Anführungszeichen drumherum setzen!

PAP oder CHAP? PAP und CHAP sind beides zwei Authentifizierungsmethoden für den PPP-Daemon. Wenn man die Wahl hat, sollte man CHAP bevorzugen, da hierbei die Daten (insbesondere das Passwort!) verschlüsselt übertragen werden. Der Aufbau der Dateien `/etc/ppp/pap-secrets` und `/etc/ppp/chap-secrets` sind nahezu identisch, von daher wurden sie oben verlinkt.

Wenn der PPP-Daemon keine Authentifizierung mit CHAP durchführen kann, wird automtisch versucht mit demselben Benutzernamen wie unter `name` angegeben eine PAP-Authentifizierung durchzuführen.

Will man die Authentifizierung mit PAP erzwingen (z.B. um den Verbindungsaufbau zu beschleunigen), setzt man den Benutzernamen hinter das Keyword `user` und kommentiert das Keyword `name` mit einem # aus (Datei `/etc/ppp/options.ippp0`).

5.3.4 Welche Daten muß ich über den Zugang kennen?

Folgende Daten sollte man kennen, die meisten sollte der ISP zur Verfügung stellen.

Protokoll

Es sollte syncPPP sein.

Telefonnummer des ISP

klar...

meine MSN

Mit welcher meiner Telefonnummern möchte/muß ich wählen, siehe 4.3 (Was ist meine MSN?).

IP-Nummern

Wenn man feste IP-Nummern hat, gibt der ISP zumindest die persönlich an, die IP-Nummer des PtP (also die des ISP) kennt deswegen noch nicht unbedingt. In diesem Fall, gibt man *irgendeine* ein (wie bei dynamischen) und läßt eine Verbindung herstellen, damit sieht man die IP-Nummer, die man dann hier fest einträgt.

Bei dynamischen IP-Nummern, trägt man *irgendwelche* ein, siehe 6 (Probleme mit dynamischen IP-Nummern).

Authentication-Typ

PAP oder CHAP. Siehe auch 5.3.3 (PAP oder CHAP?).

Username, Passwort

klar ...

Nameserver

Sollte man vom ISP mitgeteilt bekommen. Ansonsten siehe

http://www.suse.de/Support/sdb/nonameserver.html

Mit folgenden weiteren Parametern, kann man die ISDN-Verbindung steuern.

Idle-Time

Nach wie vielen Sekunden Inaktivität soll aufgelegt werden. Will man dieses Feature abstellen, kann man die Zeit auf `0` stellen.

Hinweis: Diese Zeitangabe ist nicht exakt.

Maximale Wählversuche

Wie oft soll gewählt werden, wenn der Gegner nicht abnimmt?

Hinweis: Diese Anzahl gilt nicht, wenn es Hardwareprobleme gibt, zieht man z.B. das ISDN-Kabel, wird unendlich oft probiert.

Hinweis: Diese Anzahl gilt nicht, wenn die Wählverbindung zustandekam, aber die PPP-Verbindung nicht aufgebaut werden konnte. Ist z.B. das Passwort falsch eingetragen, wird immer wieder eine Verbindung aufgebaut, solange Pakete verschickt werden.

einkommende Telefonnumern

In diesem Fall soll keiner die Verbindung von außen aufbauen dürfen, deshalb sollte man keine eingehende Telefonnummer angeben und die Option `secure` bzw. `Nur angegebene Nummern erlaubt` aktivieren.

Callback

mehr dazu siehe in

/usr/doc/packages/doc/rc.config.i4l.add

5.3.5 PPP bei S.u.S.E. einrichten

Die Konfiguration wird in der Datei `/etc/rc.config` eingetragen (außer Routing), dies kann manuell oder über YaST geschehen.

Hinweis: SuSE/YaST richtet als Authentifizierungsmethode lediglich PAP ein, wer CHAP braucht, siehe 5.3.3 (PAP oder CHAP?).

Konfiguration mit YaST:

- Menüpunkt `Administration des Systems`, `Netzwerk konfigurieren` und `Netzwerk Grundkonfiguration` auswählen.
- Es erscheint eine Maske der konfigurierten Netzdevices, wähle ein freies aus (sofern es nicht schon `ippp0` gibt. Mit `F5` wählt man den Netzwerktyp aus, hier also `ISDN SyncPPP`.
- Mit `F6` vergibt man die IP-Nummern (siehe 6.2 (Welche IP-Nummer setze ich denn eigentlich?) und kann auch direkt das Default-Gateway angeben (siehe 7 (Routing)).
- Mit `F8` werden nun die ISDN-relevanten Daten eingetragen. Nachdem man das Device aktiviert hat, kann man es in der ISDN-Maske direkt hochfahren mit: `Starten`.

Damit sind die rc.config-Variablen, die PPP-Options, die Passwortdatei und das Routing angepaßt.

manuelle Konfiguration: Durch folgende Variablen in `/etc/rc.config` wird eine syncPPP-Verbindung gesteuert, hier als echtes Bsp. (mit `_0`):

```
IPADDR_2="1.1.1.1"
NETDEV_2="ippp0"
IFCONFIG_2="1.1.1.1 pointopoint 193.102.150.13 up"
I4L_IDLETIME_2="60"
I4L_DIALMAX_2="5"
I4L_LOCALMSN_2="7417559"
I4L_REMOTE_OUT_2="52145922"
I4L_REMOTE_IN_2=""
I4L_ENCAP_2="syncppp"
I4L_SECURE_2="on"
```

Erklärung: die Bedeutung der Felder ist oben angegeben, in der `/etc/rc.config` sind auch vor den Feldern entsprechende Kommentare.

Es können beliebig viele Netzdevices definiert werden (auch wenn per Default nur 4 angegeben sind), die jeweils durch eine Extension, z.B. `_2` gekennzeichnet werden. Dabei müssen nicht alle aktiv sein. Welche aktiviert werden sollen, wird in der Variablen `NETCONFIG` gesteuert, sollen z.B. `_0` und `_2` aktiv sein, setzt man: `NETCONFIG="_0 _2"`.

Als nächstes muß der ipppd konfiguriert werden, erstelle eine Datei `/etc/ppp/options.ippp0` (siehe 5.3.2 (PPP-Optionen)) am besten in dem Du `/etc/ppp/ioptions.YaST` kopierst. In der Optionsdatei, setzte den

Usernamen und prüfe, ob das Device stimmt. Dann trägst Du das Passwort passend zum Usernamen in `/etc/ppp/pap-secrets` ein.

Zum manuellen Starten, siehe: 5.3.2 (ipppd starten)

5.4 Probleme beim Verbindungsaufbau, Troubleshooting.

Checkliste:

1. Wurde der ipppd überhaupt gestartet? Kontrolliere mit `ps ax|grep ipppd` ob einer läuft bzw. wieviele laufen.

   Kontrolliere in `/var/log/messages` die Startmeldungen, z.B.:

   ```
   syslog: info: no CHAP secret entry for this user!
   ipppd[536]: Found 1 devices: /dev/ippp0,
   ipppd[540]: ipppd i2.2.9 (isdn4linux version of pppd by MH) started
   ipppd[540]: init_unit: 0
   ipppd[540]: Connect[0]: /dev/ippp0, fd: 8
   ```

2. Stimmen die Benutzerdaten? Der ipppd prüft schon beim Start, mit welchem Usernamen gearbeitet wird (`user suse`), zu diesem Namen wird das entprechende Passwort gelesen. Klappt das nicht, wird eine Meldung ausgegeben, z.B.

   ```
   Apr  9 20:32:17 glen syslog: info: no PAP secret entry for this user!
   ```

   In diesem Fall wird eine Einwahl mittels PAP nicht funktionieren, die 12 Pfennige kann man sich also sparen. Ursache ist meist ein Tippfehler beim Benutzernamen oder falsche Permisssions der pap-secrets.

   Analoges gilt für CHAP.

3. Wird überhaupt eine Verbindung aufgebaut? Sobald die Gegenstelle abnimmt (und damit Kosten entstehen) erscheint eine `connect`-Meldung.

   Wird keine Verbindung aufgebaut, gibt es vermutlich eine `cause`-Meldung. Siehe: 4.4 (Cause Meldungen).

   Erscheinen nur `dialing`-Meldungen, aber sonst nichts, liegt es an der Hardware oder am Hardware-Modul, siehe 3.4 (Hardware testen) und 11 (Installation).

4. Klappt die Einwahl? Bei Erfolgreicher Einwahl erscheinen Meldungen über die IP-Nummern, z.B.:

   ```
   ipppd[540]: local  IP address 149.228.142.59
   ipppd[540]: remote IP address 193.102.150.13
   ```

   Sieht man diese Meldungen, dann Glückwunsch! Der Gegner wird ab jetzt zum Partner.

5. `select: Bad file number` Direkt nach der Ausgabe der IP-Nummern ercheint:

   ```
   ipppd[353]: select: Bad file number
   ipppd[353]: Couldn't restore device fd flags: Bad file number
   ipppd[353]: Exit.
   ```

Was hat es damit auf sich? Zunächst einmal: die Verbindung ist trotz allem aufgebaut.

Ursache: der ipppd startet beim (Dis-) Connect das Script `/etc/ppp/ip-up` (`ip-down`). Aufgrund eines kleinen Fehlers im offiziellen ipppd (behoben in der CVS-Version und ab S.u.S.E. 5.2) ist die Abprüfung auf Ausführbarkeit fehlerhaft, woraufhin trotzdem versucht wird das Script zu starten. Nach der Ferhlermeldung passiert genau nichts.

Allerdings sollte die Meldung trotzdem beachtet werden, denn da wir dynamische IP-Nummer benutzen, muß das Routing angepaßt werden, was genau über diese Scripte geschehen soll, die hier nicht vorhanden sind.

6. **Die Einwahl klappt nicht:** Wenn die Einwahl nicht klappt, sieht man in `/var/log/messages` meist nur, daß die Gegenstelle aufgelegt hat, um den Grund dafür zu ermitteln, braucht man mehr Meldungen vom LCP. Dazu trägt man in `/etc/ppp/ioptions` folgendes ein:

```
# Set 'debug' to create a lot of information in /var/log/messages
debug
# Set '+pwlog' for logging passwords in /var/log/messages
#+pwlog
```

und startet den ipppd neu. Durch `debug` werden die LCP-Messages ausgegeben, durch `+pwlog` kann man sich zusätzlich das verschickte Passwort angeben lassen - letzteres ist nur empfohlen, wenn ansonsten alles richtig scheint, denn wenn jemand fremndes Zugriff auf `/var/log/messages` bekommt...

Häufige Ursachen:

- Username/Passwort falsch: in diesem Fall wird etwas in dieser Art protokolliert,

```
ipppd[10314]: sent [0][PAP AuthReq id=0x1 user="kfr" password="falsch"]
ipppd[10314]: rcvd [0][PAP AuthNak id=0x1msg=""]
ipppd[10314]: Remote message:
ipppd[10314]: PAP authentication failed
```

  wobei es richtig so aussehen sollte:

```
ipppd[7840]: sent [0][PAP AuthReq id=0x1 user="kfr" password="isdnworkshop"]
ipppd[7840]: rcvd [0][PAP AuthAck id=0x1msg=""]
ipppd[7840]: Remote message:
ipppd[7840]: bundle, he: 0 we: 0
```

  Siehe auch 5.3.3 (Sonderzeichen im Usernamen oder Passwort).

- LCP-Messages werden nicht beantwortet: Normalerweise LCP-Messages gesendet und empfangen um das Handshaking durchzuführen (send, rcvd):

```
ipppd[10314]: sent [0][LCP ConfReq id=0x1 <mru 1524> <magic0x93ade903>]
ipppd[10314]: rcvd [0][LCP ConfReq id=0x1 <mru 1524> <auth pap>
<MPdiscr: 0x3 [ 00 c0 7b 70 d5 fa ]>]
```

  Wenn die Gegenseite nicht antwortet, kann es sein, daß sie nicht schnell genug hochkommt (`lcp-restart` erhöhen), oder kein (sync-) PPP-Daemon dort läuft. Ist dies nicht nur ein temporäres Problem, ist entweder die Nummer falsch, oder der ISP bietet tatsächlich kein syncPPP an.

  Im letzteren Fall muß man asyncPPP fahren, siehe

  http://www.suse.de/Support/sdb/ppp_async.html

- Noch während der LCP-Messages legt die Gegenstelle auf. Hierbei sollte man am Protokoll erkennen welche Optionen angefordert oder abgewiesen werden. Danach bleibt einem nur der mühsame Weg, diese Optionen zu setzen/deaktivieren, siehe Bsp-Optionsfile und `man ipppd`.
- `peer refused to authenticate` Man hat selbst `+pap` oder `+chap` angegeben. Ein häufiges Mißverständnis: diese Optionen geben, an, daß man selber der Authentication-Server sein möchte, nicht daß man PAP oder CHAP benutzen möchte; letzteres geschieht nur implizit durch Angabe `user`, `name` und den Eintragungen in `pap-secrets` bzw. `chap-secrets`.

7. **Die Einwahl klappt, weitere Tests:**
   - Zunächst vergleiche man die Ausgabe des ipppd mit der Ausgabe von `ifconfig`. Die IP-Nummern müssen übereinstimmen (und gegenüber der Grundeinstellung verändert sein).
   - Ist das Routing richtig gesetzt? Prüfe `route -n`. Siehe 7 (Routing). Es muß eine Hostroute für den PtP-Partner gesetzt sein.
   - Versuche die Gegenstelle anzupingen, z.B. `ping 193.102.150.13`.
   - Warte, bis die Verbindung automatisch abbricht und prüfe die Routingtabelle erneut.
   - beobachte, ob wieder automatische gewählt wird.

5.5 Übung: syncPPP-Verbindung herstellen

Ziel: PPP-Verbindung aufbauen und testen (kein Routing)

1. Stelle eine Verbindung zu einem syncPPP-Server her. Wenn Du keinen anderen kennst, benutze den S.u.S.E. ISDN-Testrechner mit folgenden Daten:
   - Protokoll: syncPPP
   - Telefon: +49 911 3206726
   - Username: suse
   - Passwort: linux
   - IP-Nummer Server: 192.168.0.1
   - IP-Nummer Client: 192.168.0.99
2. Gehe die Checkliste durch und führe die dortigen Tests aus, siehe 5.4 (syncPPP Troubleshooting)
3. Prüfe die IP-Nummer(n) des Servers, wenn diese fest ist, ändere die Konfiguration entsprechend.

6 Probleme mit dynamischen IP-Nummern

Was sind dynamische IP-Nummern?

IP-Nummern sind knapp und daher teuer. Die Provider versuchen deshalb IP-Nummern einzusparen, indem sie sich nur (mindestens) so viele IP-Nummern zuweisen lassen, wie sich gleichzeitig bei Ihnen einwählen können. Die Anzahl der potenziellen Rechner, die sich einwählen könnten ist aber höher, daher kann nicht mehr für jeden Rechner eine IP-Nummer fest zugeordnet werden.

Der Trick besteht also darin, daß auf eine feste Zuordnung Rechner - IP-Nummer verzichtet wird und stattdessen bei jedem Verbindungsaufbau aus einem freien Pool eine ausgewählt wird, die dem Client mitgeteilt wird. Diese Technik nur beim PPP-Protokoll benutzt werden, nicht bei rawip.

Diese Methode ist prima, wenn man nur eine Arbeitstation hat und Session-orientiert arbeitet: Verbindung aufbauen, surfen, surfen, Mails austauschen, surfen und schließlich Verbindung abbauen.

Will man nur ein klein wenig mehr (transparenten Internetzugriff), stellt sich schnell heraus, daß das Internetkonzept und dynamische IP-Nummern nicht zusammenpassen.

Folgende Punkte sind für einen transparenten Internetzugriff wünschenswert:

1. Dial-on-demand: es wird nicht manuell entschieden, daß eine Verbindung auf- oder abgebaut werden soll (wer soll das in einem größeren Netz machen?). Die Wählleitung wird automatisch aufgebaut, sobald Pakete vorhanden sind, die nicht im lokalen Netz zugestellt werden können.
2. Automatischer Verbindungsabbau, wenn eine gewisse Zeit keine Pakete über die Leitung gehen.
3. Pausen verursachen keine Kosten, wenn keine Daten über die Leitung gehen, liest man z.B. eine etwas längere Web-Seite, braucht die Wählleitung nicht aufgebaut zu bleiben.
4. Verhindern, daß vergessen wird aufzulegen. (Es blinkt und klackt ja nicht ...)

Dieses läßt sich mit ISDN wunderbar lösen, vor allem deshalb, weil der Verbindungsaufbau im Gegensatz zu einem Modem sehr schnell geht (wenige Sekunden).

Folgende Punkte sind bei dynamischen IP-Nummern nicht realisierbar:

1. Server-Funktionalität: die IP-Nummer des Rechners ist für einen anderen Rechner im Internet nicht bestimmbar. Außderdem wird der Provider vermutlich nicht selbst eine Wählverbindung aufbauen wollen und können - zumindest nicht bei diesen kostengünstigen Verträgen.
2. Mails können nicht direkt zum eigentlichen Mailserver durchgestellt werden (an welche IP-Nummer sollten sie denn?), sondern müssen bei einem Provider abgelegt werde, von dem sie vom IZG abgeholt werden.
3. Das **Offene-Sockets-Problem**: Halten einer logischen Verbindung über die Verbindungsunterbrechnung hinaus ist nicht möglich. Bsp: man loggt sich per Telnet bei in seiner Arbeitsstelle ein, nach Inaktivität wird aufgelegt, drückt man nun eine Taste, wird die Verbindung automatisch wieder hergestellt, aber man bekommt eine andere IP-Nummer zugewiesen. Die Socket-Verbindung zwischen eigenen Rechner und Arbeitgeber ist damit ungültig geworden, da für einen Socket u.a. Quell- und Ziel-IP-Nummern wichtig sind.

   Die gleiche Problematik stellt sich bei WWW oder FTP-Verbindungen, die unterbrochen werden.

Sehr wohl aber ist man genauso wie sonst auch nicht gegen Angriffe aus dem Internet geschützt. Ein Hacker kann nur nicht voraussagen, welchen Rechner er angreift, er sucht sich halt nur zufällig eine IP-Nummer aus (oder belauscht eine Verbindung) und kann den Rechner angreifen. Der Vorteil liegt allerdings darin, daß der Hacker weniger Zeit hat, da die Verbindung abgebaut und mit einer neuen IP-Nummer aufgebaut wird, zwischen denen erstmal kein Zusammenhang zu erkennen ist. Als wirksamen Schutz reicht dies aber nicht aus.

Die Probleme: Aus dem **Offene-Sockets-Problem** ergeben sich zwei Punkte, die bei einem IZG mit dynamsichen IP-Nummer beachtet werden müssen:

1. Anfragen laufen in's Leer: ein Web-Browser hat einen zum Web- oder Proxy-Server offen, nach dem Re-Connect ist dieser ungültig, aber der Browser hat keine Möglichkeit dies zu erkennen. Abhilfe schafft es hier, auch `Abbruch` und `Reload` zu drücken.
2. Die Sockets bleiben offen (auch nach Beendigung des Client-Programms), es werden immer wieder Pakete darüber geschickt, bis ein Timeout (etwa 20 Minuten) abläuft. In dieser Zeit wird **ständig eine Verbindung aufgebaut** bzw. bleibt bestehen.

Abhilfe schafft hier zum einen, daß man den Client nicht erlaubt direkt in das Internet eine Verbindung aufzubauen (über Masquerading), sondern nur über Proxies (siehe 10.2 (Squid). Aber auch diese Methode ist nicht zuverlässig.

6.1 Der RST-provoking mode

Wirkliche Abhilfe schafft nur die Aktivierung des **RST-provoking mode**. Dabei wird bei dem Paket die Quell-IP-Nummer ausgetauscht gegen die jetzt aktuelle dynamsiche IP-Nummer, was bewirkt, daß beide Seiten diesen Socket schließen.

Diese Modus ist leider noch nicht in den offiziellen Kernel gekommen. Den Patch von **Erik Corry** findet man hier:

http://www.image.dk/~ehcorry/linux/

Er ist für Kernel der Version bis 2.0.33 passend, ab Version 2.0.34 wird er vermutlich im Standard-Kernel sein. Im Standrdkernel von S.u.S.E. Linux 5.2 (und im Quellpaket `lx_suse` ist dieser Pacth schon enthalten. (Offen: 2.1?)

Zur Aktivierung gibt man das Kommando:

```
echo 7 > /proc/sys/net/ipv4/ip_dynaddr
```

(Oder nur `5`, für den quiet-Mode). Bei Erfolg sieht man in `/var/log/messages` Meldungen der folgenden Art:

```
ip_rewrite_addrs(): shifting saddr from 1.1.1.1 to 149.228.142.50 (state 2)
```

Aktivierung bei S.u.S.E.:

Trage in `/sbin/init.d/i4l_hardware` vor dem Start des isdnlog folgende Zeilen ein:

```
test -z "$I4L_DYNIP" ||
    echo 7 > /proc/sys/net/ipv4/ip_dynaddr
```

(das wird vermutlich bei S.u.S.E. Linux später als 5.2 der Fall sein) und trage in `/etc/rc.config` ein:

```
I4L_DYNIP="yes"
```

6.2 Welche IP-Nummer setze ich denn eigentlich?

Der Provider stellt nur dynamische IP-Nummern zur Verfügung, während der Konfiguration von i4l werde ich aber nach IP-Nummer gefragt - welche IP-Nummer soll ich denn da angeben?

i4l arbeitet mit einer transparenten Netzanbindung, d.h. logisch gesehen ist die Verbindung immer aktiv, auch wenn noch garnicht gewählt wurde und keine dynamischen IP-Nummern ermittelt werden konnten. Um dieses Pseudo-Netzwerk zu konfigurieren müssen aber zwangsläufig IP-Nummern angegeben werden.

Es empfiehlt sich daher, eine Pseudo-IP-Nummer zu benutzen, z.B. dieselbe, die man auch für seine Ethernetanbindung benutzt. Das ist möglich, da die PPP-Verbindung als `pointopoint`-Verbindung (beim `ifconfig`)

konfiguriert wurde, dies ist ein spezieller Modus, durch den der Kernel weiß, daß hier nur eine Verbindung zwischen zwei Punkten stattfindet. Warum Point-To-Point (**PtP**) als **pointopoint** angegeben wird, weiß ich auch nicht ....

Um keinen Konflikt mit offiziellen IP-Nummern zu provozieren, sollte man eine aus dem privaten Bereich wählen, z.B. `192.168.1.1`.

Falls man bei T-Online angeschlossen ist oder dies plant: Benutze nicht `192.168.0.*`, darüber werden z.T. interne Dienste wie Cept abgehandelt.

7 Routing

7.1 Was ist Routing?

In einem lokalen Netzwerk ist das Leben einfach: wenn ein TCP-IP Paket zu einem anderen Rechner gesendet werden soll, wird dieses auf dem Ethernet verschickt.

Ist der Rechner am Internet oder in einem grösseren Netzwerk (WAN) angeschlossen, ist die Aufgabe schon etwas schwieriger, denn wenn der Ziel-Rechner (bzw. Ziel-IP-Nummer) nicht im lokalen Ethernet erreichbar ist, so muss dem Kernel gesagt werden, daß alle nicht lokal zustellbaren Pakete, freundlicherweise von einem Gatewayrechner weitergeleitet werden.

Komplizierter ist es, wenn der betreffende Rechner selbst ein Gatewayrechner ist und mehrere Netzdevices (Ethernetkarten, Modems, ISDN-Karten etc.) zur Verfügung hat und jeweils über diese Devices unterschiedliche Rechner/Netze erreichbar sind. Das ist die Aufgabe vom Routing:

> `Für jede IP-Nummer muß definiert werden, auf welchem Weg (Route) diese erreicht werden kann.`

Man unterscheidet folgende Typen: (die Beispiele werden unter konkretisiert)

Netzrouten

Hier wird angeben, wie ein komplettes Netz erreichbar ist. Beispiel für ein lokales Ethernet:

> `Bsp 1: Das Netz 192.168.1.0 mit der Maske 255.255.255.0 ist über das Device eth0 erreichbar.`

Hostrouten

Man definiert, wie ein einzelner Rechner erreichbar ist. Beispiel für eine syncPPP Verbindung:

> `Bsp 2: Der Rechner 192.168.0.1 ist über das Device ippp0 erreichbar.`

Default-Route

Im Internet gibt es recht viele IP-Nummern - es ist daher mühsam und langweilig für alle einzelnen IP-Nummern oder Netze einzelne Routing-Einträge zu machen. Daher gibt es die Möglichkeit zu sagen:

> `Bsp 3: Alle IP-Nummern, für die keine spezielle Regel vorhanden ist, schicke an den Rechner mit der IP-Nummer 192.168.0.1.`

Man beachte: es macht i.A. keinen Sinn, mehr als eine Default-Route anzugeben.

7.2 Wie konfiguriert man das Routing?

Die Routingeinträge werden dem Kernel zur Laufzeit mit dem Kommando `route` mitgeteilt (und wieder entzogen).

7.2.1 S.u.S.E. Methode

Bei S.u.S.E. können die Routingeinträge fest in die Datei `/etc/route.conf` eingetragen werden, die beim Booten oder durch einen Runlevelwechsel vom Script `/sbin/init.d/route` ausgewertet wird.

Die Einträge für die obigen Beispiele sehen so aus:

```
# Bsp 1:
192.168.1.0     0.0.0.0     255.255.255.0   eth0
# Bsp 2:
192.168.0.1     0.0.0.0     255.255.255.255 ippp0
# Bsp 3:
default         192.168.0.1
```

Die **1. Spalte** gibt das Ziel an, also das Netz, die IP-Nummer, oder das Schlüßelwort `default`. In der **3. Spalte** steht die zugehörige Netzmaske (falls notwendig).

In der **2. Spalte** steht der Gatewayrechner, an den die Anfragen geschickt werden sollen.

In der **4. Spalte** steht das zu verwendene Device.

Hier sieht man auch in der 3. Zeile, daß bei Verwendung eines Gatewayrechners die Angabe des Devices nicht nötig ist, da sie selbstständig ermittelt wird.

Allerdings muß (in diesem Beispiel) die Hostroute auf `192.168.0.1` definiert sein, bevor man sie zum Setzen der Defaultroute nutzen kann. Merke: **Die Reihenfolge ist wichtig.**

Manuelles Setzen und Löschen der Routingtabelle:

```
/sbin/init.d/route start
/sbin/init.d/route stop
```

7.2.2 Manuelle Methode

```
# Bsp 1:
route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
# Bsp 2:
route add -host 192.168.0.1 dev ippp0
# Bsp 3:
route add default gw 192.168.0.1
```

Mehr Infos: `man route`.

7.2.3 Löschen von Routing-Einträgen

Routing-Einträge können zum einem direkt gelöscht werden, sie werden aber auch automatisch gelöscht, wenn das zugrundeliegende Netzdevice gelöscht oder umkonfiguriert wird.

Dies hat in diesem Zusammenhang einen ungewünschten Nebeneffekt. Der `ipppd` baut die Verbindung auf und bekommt eine neue IP-Nummer vom Server zugewiesen, wobei selbstständig eine neue Hostroute auf die IP-Nummer des Gegners eingerichtet wird.

Allerdings wird eine ev. vorhandene Defaultroute über dieses Device gelöscht.

Durch die PPP-Option `defaultroute` könnte man sich automatisch wieder eine Anlegen lassen. Allerdings ist diese Methode nicht sehr flexibel (vielleicht will man ja doch keine Defaultroute) und man hätte hiermit keine Möglichkeit zu steuern, wie sich beim Verbindungsabbau verhalten werden soll. Daher wird beim Verbindungauf- und abbau jeweils ein Script gestartet, siehe 7.3 (Kontrollieren der Routingtabelle beim Verbindungsauf- und abbau).

7.3 Kontrollieren der Routingtabelle beim Verbindungsauf- und abbau (/etc/ppp/ip-up)

Der `ipppd` bietet die einfache Möglichkeit beim Verbindungsaufbau das Script `/etc/ppp/ip-up` und beim Abbau `/etc/ppp/ip-down` zu Starten, wobei jeweils die folgenden Parameter über den neuen Zustand übergeben werden:

- $1: Interface
- $2: Device
- $3: Speed (nur aus Kompatibilitätsgründen)
- $4: lokale IP-Nummer
- $5: IP-Nummer des Gegners

Durch Installation geeigneter Scripte kann also die Default-Route neu gesetzt werden. Die Scripte könnten jeweils so aussehen:

```
#!/bin/sh
/sbin/route add default gw $5
```

Bei S.u.S.E. wird ein Script `/etc/ppp/ip-up` welches für den *hausgebrauch* ausreicht. Die Routen werden aufgrund der Konfigurationsdateien gesetzt und wieder hergestellt. Weitere Kommandos können vom Administrator eingefügt werden (z.B. Mails verschicken).

Das Script `ip-down` ist ein symbolischer Link auf `ip-up`, so daß man nur eine Datei zu verwalten hat.

7.3.1 Was macht das Script ip-up/ip-down?

Es wird geprüft. ob das Interface ippp? ist, sollte also bei Analog-PPP nicht stören, wer dort etwas eintragen will, sollte die Stelle leicht finden.

Wenn es als `ip-up` aufgerufen wird (also nach dem Verbindungsaufbau), wird eine Default-Route auf die gerade zugewiesene IP-Nummer gesetzt.

Wenn es als `ip-up` aufgerufen wird (also nach dem Verbindungsabbau), dann wird das Interface gelöscht. Das Interface wird wie in `/etc/rc.config` wieder neu angelegt, es wird also wieder auf die ursprünglichen IP-Nummer gesetzt. Nach den Angaben in `/etc/route.conf` werden die Routingeinträge für dieses Device neu eingerichtet. Somit ist dial-on-demand wieder möglich. Ist dort keine Defaultroute angegeben, wird auch keine gesetzt.

Ich möchte aber kein dial-on-demand In der `/etc/route.conf` (bzw. in YaST) wird keine Default-Route (Default-Gateway) angeben, dadurch existiert nur während einer Verbindung eine Default-Route, diese wird beim Verbindungsabbau gelöcht und nicht neu angelegt. Die Verbindung kann dann manuell (oder durch ein Script) durch `isdnctrl dial ippp0` aufgebaut werden (oder durch manuelles setzen der Default-Route).

Dadurch kann z.B. auch erreicht werden, dass mit verschiedenen Providern gearbeitet wird, in dem Fall muss man ja sowieso entscheiden, welche Verbindung nun hochgefahren werden soll, z.B. `isdnctrl dial ippp17`

7.4 Übung: Kontrolliere die IP-Nummer und die Routing-Tabelle

1. `/var/log/messages` überwachen Siehe 2 (Betrachte messages)
2. Prüfe `ip-up` und `ip-down`

```
glen:/root # ls -la /etc/ppp/ip-*
lrwxrwxrwx   1 root     root            5 Mar 20 10:16 /etc/ppp/ip-down -> ip-up
-rwxr-xr-x   1 root     root         1813 Mar 24 23:03 /etc/ppp/ip-up
```

   Siehe 11 (Installation)

3. Prüfe IP-Nummern und die Routingtabelle **vor** einer Verbindung

```
glen:/root # ifconfig ippp0
ippp0     Link encap:Point-Point Protocol
inet addr:192.168.0.99  P-t-P:192.168.0.1  Mask:255.0.0.0
UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0
TX packets:0 errors:0 dropped:0 overruns:0


glen:/root # route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.1     0.0.0.0         255.255.255.255 UH    0      0        0 ippp0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        2 lo
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 ippp0
```

4. Verbindung initiieren Man kann entweder ein Pakete verschicken (z.B. `ping 141.1.1.1` oder das Wählen direkt verlangen `isdnctrl dial ippp0`

   Als Beispiel bekommen wir die IP-Nummer `1.2.3.4` zugewiesen, der Gegner habe die IP-Nummer `5.6.7.8` (siehe messages).

5. Prüfe IP-Nummer und die Routingtabelle **während** einer Verbindung

```
glen:/root # ifconfig ippp0
ippp0     Link encap:Point-Point Protocol
inet addr:1.2.3.4  P-t-P:5.6.7.8  Mask:255.0.0.0
UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
RX packets:2 errors:0 dropped:0 overruns:0
TX packets:3 errors:0 dropped:0 overruns:0
```

```
glen:/root # route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
5.6.7.8         0.0.0.0         255.255.255.255 UH    0      0        0 ippp0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        2 lo
0.0.0.0         5.6.7.8         0.0.0.0         UG    0      0        0 ippp0
```

6. Wir gehen in die große weite Welt: Bestimme eine existierende IP-Nummer, die einzige, die ich mir merken kann ist die des DNS-Server von ECRC: `traceroute -n 141.1.1.1`. Man beachte, daß wir noch keinen DNS-Servive benutzen können, daher `-n`.

7. Timeout abwarten bis aufgelegt wird/ betrachte `/var/log/messages`, z.B.:

```
kernel: isdn_net: local hangup ippp0
kernel: ippp0: Chargesum is 0
isdnlog: Apr 03 09:20:49   tei 70 calling Eunet-N with KfrI I  Normal call clearing (User)
ipppd[135]: Modem hangup
ipppd[135]: Connection terminated.
ipppd[135]: taking down PHASE_DEAD link 0, linkunit: 0
ipppd[135]: sent [0][LCP TermReq id=0x2 6c 69 6e 6b 20 63 6 c 6f 73 65 64]
ipppd[135]: LCP is down
ipppd[135]: link 0 closed , linkunit: 0
ipppd[135]: reinit_unit: 0
ipppd[135]: Connect[0]: /dev/ippp0, fd: 6
```

8. IP-Nummern und Routing prüfen sie müssen jetzt wieder genausogesetzt sein, wie **vor** dem Verbindungsaufbau.

8 IP-Nummern Auflösung (DNS)

Bekanntlich werden Rechner im Internet über die IP-Nummern angesprochen. Niemand möchte sich aber die IP-Nummern direkt merken, praktischer ist es, Namen zu verwenden, z.B. `www.franken.de`. Aber nicht nur für die bessere Merkbarkeit sind diese Namen wichtig, sondern sie dienen auch als Variable, deren Inhalt sich verändern kann. Wenn ein wichtiger Server eine neue IP-Nummer bekommt (z.B. durch Umzug oder Providerwechsel), so wird der Name einfach auf die neue IP-Nummer aufgelöst.

Genauso wichtig (und das wird gerne vergessen) wie die Auflösung von Namen in IP-Nummern ist der umgekehrte Fall, also IP-Nummer in einen Rechnernamen auflösen.

Diese umgekehrte Auflösung ist oft diejenige, die Probleme im lokalen Netzwerk (also ungewollte Verbindungen) macht, denn viele Services nutzen diese Möglichkeit zur Verifikation bei einer einkommenden Verbindung, denn in den Regeln wer was machen darf, werden meist Rechnernamen anstatt IP-Nummern verwendet. Über das Netzwerk ist aber zunächst nur die IP-Nummer sichtbar und muß also in einen Namen aufgelöst werden.

Es gibt zwei wichtige Methode zur Namensauflösung, die gleichzeitig benutzt werden können (und müssen):

8.1 feste IP-Nummern Auflösung über /etc/hosts

Alle bekannten IP-Nummer werden fest in einer Datei gespeichert, die der Adminsitrator manuell pflegen (oder kopieren) muß.

In der Datei `/etc/hosts` werden alle Rechnernamen und IP-Nummern fest eingetragen.

Beispiel: In der Domain `isdnworkshop.de` gibt es die Rechner Asterix (`192.168.1.1`) und Obelix (`192.168.1.2`). Dann sieht die Datei so aus:

```
# IP        FQN                      Kurzname

192.168.1.1 Asterix.isdnworkshop.de Asterix
192.168.1.2 Obelix.isdnworkshop.de Obelix
```

8.2 dynamische IP-Nummern Auflösung mit DNS

Es wird schnell ersichtlich, daß eine feste Auflösung über eine Datei, die ständig aktuell auf jedem Rechner installiert sein muß das Internet nicht funktionieren würde. Die feste Auflösung kann nur in einem übersichtlichen lokalen Netz benutzt werden.

DNS (Domain Name Service) dient ebenfalls zum Auflösen von Rechnernamen in eine IP-Nummer und umgekehrt. Der Unterschied liegt darin, daß es ein Internet-Service ist, den man auf Anforderung abfragen kann. Es gibt sehr viele DNS-Server im Internet, wobei es eine hierarchische Struktur gibt, die sich an den Domainnamen orientiert. Jeder DNS-Server ist für eine Sub-(Sub-....) Domain zuständig. Beim Abfragen *hangelt* man sich von den Root-Servern herunter, bis man den Server gefunden hat, der die Anfrage tatsächlich beantworten kann.

Das Einrichten eines DNS-Server soll an anderer Stelle beschrieben werden, z.B. im *DNS HOWTO*.

Für unsere Zwecke reicht es zu wissen, wie der Service aktiviert wird und wo man einstellt, welches der Nameserver ist.

8.3 Konfiguration der Namensauflösung

Es ist wie gesagt durchaus sinnvoll beide Methoden der Namensauflösung zu kombinieren. Wichtig ist hier, daß auch ohne Internetverbindung lokal gearbeitet werden kann. Üblicherweise werden die lokalen Rechner (mindestens der eigene) über die `/etc/hosts` aufgelöst, alle nicht bekannten Anfragen werden dann über den Nameserver beim ISP aufgelöst.

Um die Namensauflösung muß sich eine Applikation nicht selber kümmern, sondern wird durch `libc`-Funktionen (z.b. `gethostbyname()` erledigt. Diese libc-Funktionen gilt es also zu konfigurieren.

Über die Datei `/etc/host.conf` wird zunächst gesteuert, welche Methoden überhaupt benutzt werden sollen und sehr wichtig auch in welcher **Reihenfolge** dies geschehen soll.

Beispiel `/etc/host.conf`:

```
order hosts bind
multi on
```

gibt an, daß zunächst in der `/etc/hosts` gesucht werden soll, bei Mißerfolg dort, soll der DNS-Server (`bind`) bemüht werden.

Wenn ein Nameserver benutzt werden soll, ist noch eine zweite Datei `/etc/resolv.conf` zu konfigurieren:

```
search isdnworkshop.de suse.de
nameserver 192.168.200.7.1
```

Die 2. Zeile sollte selbsterklärend sein, in der ersten wird eine sogenannte Searchlist angegeben, diese ist nur dann von Bedeutung, wenn ein Rechnername ohne vollständige Domain versucht wird aufzulösen. Beispiel: es wird nach einem Rechner `Goedel` gesucht, den der Nameserver nicht kennt, dann wird zunächst `isdnworkshop.de` angehängt und damit versucht einen Rechner `Goedel.isdnworkshop.de` zu finden; ist auch das nicht erfolgreich, wird nach `Goedel.suse.de` gesucht.

Änderungen an diesen beiden Dateien sind sofort wirksam.

8.3.1 Namensauflösung bei S.u.S.E.

Setze die Variablen in `/etc/rc.config`, für obiges Beispiel:

```
SEARCHLIST="isdnworkshop.de suse.de"
NAMESERVER="192.168.200.7.1"
```

8.4 Probleme mit der Namensauflösung

Probleme bei der Namensauflösung erkennt man schnell an seiner Telefonrechnung ;-(

Ein Beispiel: eine Benutzer macht im lokalen Netz ein Telnet von der IP-Nummer `192.168.1.2` auf den IZG `192.168.1.1`. Der Server prüft vor dem eigentlichen Start des Telnet-Daemons, welche IP-Nummer reinkommt (Stichwort TCP-Wrapper), da diese Nummer nicht aufgelöst werden kann, wird der Nameserver befragt, dieser ist beim ISP, eine Verbindung wird automatisch aufgebaut. Ergebnis: der Telnet braucht nicht nur etwa eine Minute bis zum Login (der DNS-Server kann diese private IP-Nummer nicht auflösen), sondern kostet auch noch 12 Pfennige.

8.4.1 Checkliste

1. Ist die eigene IP-Nummer in der `/etc/hosts` eingetragen?
2. Sind alle Rechner des lokalen Netzwerks in der `/etc/hosts` eingetragen?
3. Ist das Paket `bind` installiert:

   ```
   +/kfr $ rpm -q bind
   bind-4.9.6-5
   ```

4. Kann der Nameserver angesprochen werden? Test:

   ```
   +/kfr $ nslookup www.suse.de
   Server:  Plato.suse.de
   Address:  192.168.100.1

   Name:    Turing.suse.de
   Addresses:  195.125.217.200, 192.168.102.3
   Aliases:  www.suse.de
   ```

5. Einen beliebigen anderen Nameserver kann man direkt testen, z.B.:

   ```
   +/kfr $ nslookup www.suse.de 141.1.1.1
   Server:  ecrc.de
   Address:  141.1.1.1
   ```

```
Non-authoritative answer:
Name:    Turing.suse.de
Address:  195.125.217.200
Aliases:  www.suse.de
```

Tips:

1. Für das gesamte Subnetz IP-Nummern und Namen in die `/etc/hosts` eintragen, auch wenn sie (noch) nicht verwendet werden. Bsp:

   ```
   192.168.1.1     Server.isdnworkshop.de Server
   192.168.1.2     Client.isdnworkshop.de Client
   192.168.1.3     Dummy.isdnworkshop.de Dummy
   192.168.1.4     Dummy.isdnworkshop.de Dummy
   192.168.1.5     Dummy.isdnworkshop.de Dummy
   ```

   u.s.w.

2. Einrichten eines eigenen DNS-Proxy-Servers. Neben der schnelleren Auflösung, werden auch die fehlerhaften Anfragen gecacht, so daß nicht so häufig eine Verbindung aufgebaut wird (Siehe 10.1 (DNS-Cache)).

9 Dial-On-Demand kontrollieren

Während der Konfiguration sollte man unbedingt das System überwachen und feststellen, wann und warum eine Verbindung aufgebaut wird. Ansonsten kann es schnell zu unerwünschten Telefonrechnungen kommen.

Man kann sich aber sicher sein, daß niemals grundlos eine Verbindung aufgebaut oder offengehalten wird. Die geschieht immer nur dann, wenn auch tatsächlich Pakete über die Leitung verschickt werden.

Es gilt also inbesondere die beteiligten Serverdienste auf dem Rechner zu überprüfen, ob Sie richtig konfiguriert wurden und ggf. die Ursachen der Verbindung aufzuspüren.

9.1 Verbindungen überwachen

Es gibt eine Vielzahl von ISDN-Statusmonitoren, der wichtigste ist `imon`; dieses Konsolenprogramm läßt sich in jeder Umgebung einsetzen, reagiert prompt und verschlingt keine Systemressourcen.

Weitere Programme sind: `xisdnload` (zeigt auch den Durchsatz), `isdnmon` und `isdnmonp`. Alle Monitore zeigen die Telefonnummer und die Art der Verbindung (ein- oder ausgehende an).

9.2 Grund der Verbindung feststellen

- Durch den Befehl `isdnctrl verbose 3` wird das i4l-Subsystem veranlasst, bei jedem Verbindungsaufbau eine Meldung in `/var/log/messages` zu schreiben, anhand der man erkennen kann, zwischen welchen IP-Nummern und Port-Nummern ein Paket verschickt wird.

  Dieses Beispiel ist eine Anfrage an den WWW-Server `www.suse.com` (Alias `goldengate`):

  ```
  Apr 10 21:05:06 glen kernel: OPEN: 1.1.1.1 -> 209.0.51.1 TCP, port: 2224 -> 80
  ```

Nachteil: man kann nicht überprüfen, warum eine Verbindung nicht abgebaut wird.

Mehr: 13.1 (SDB: ungewollte Verbindungen)

- `tcpdump` (Paket `tcpdump`) ist ein Paketsniffer, der alle Pakete auf einem Netzdevice mitschneidet. Die Ausgabe des Programmes ist leider nicht sehr menschenfreundlich, aber zumindest die verwendeten IP-Nummer und Port-Nummern werden sichtbar gemacht.

  Dieses Beispiel ist eine Anfrage an den WWW-Server `www.suse.com` (Alias `goldengate`):

```
glen:/root # tcpdump -i ippp0
tcpdump: listening on ippp0
21:05:39.382188 pec-30.au1.n.uunet.de.2230 > goldengate.suse.com.www:
    S 1384488919:1384488919(0) win 512 <mss 1460>
21:05:39.642188 goldengate.suse.com.www > pec-30.au1.n.uunet.de.2230:
    S 3326089293:3326089293(0) ack 1384488920 win 32736 <mss 1460>
21:05:39.642188 pec-30.au1.n.uunet.de.2230 > goldengate.suse.com.www:
    . ack 1 win 32120 (DF)
```

  Nachteil: bei Verwendung dynamischer IP-Nummern wird durch den PPP-Daemon das Interface `ippp0` neu angelegt. `tcpdump` zeigt nach dem Neuanlegen keine Daten mehr an und muß abgebrochen und neu gestartet werden.

9.3 Verbindungen auswerten

Das Programm `isdnlog` läuft im Hintergrund und horcht ständig auf dem D-Kanal mit, alle Aktivitäten werden zum einen in `/var/log/messages` geloggt, zum anderen in die Log-Datei `/var/log/isdn.log` protokolliert.

Mit dem Tool `isdnrep` kann man diese Datei wiederum zu einem späteren Zeitpunkt aufrufen. Es gibt eine Vielzahl von Parametern, hier dir wichtigsten:

- `isdnrep`: alle Verbindungen des heutigen Tages
- `isdnrep -a`: alle protokollierten Verbindungen
- `isdnrep -t01/04/98-03/04/98`: alle Verbindungen vom 1. bis 3. April 1998

Mehr Infos in

/usr/doc/packages/i4l/isdnlog/README

bzw. im Quellpaket.

9.4 Dial-On-Demand an- und ausstellen

Das i4l-Subsystem ist (wenn es denn einmal gestartet wurde) nicht dafür vorgesehen, daß Verbindungen nur manuell gestartet werden. Man könnte das Konzept bei i4l also auch so formulieren: wenn es gestartet ist, besteht ständig eine Verbindung, die aber automatisch gekappt wird, wenn nichts passiert.

Wer es dennoch machen will, der entferne einfach die Default-Route. In diesen Fall wird nur noch dann eine Verbindung aufgegebaut, wenn ein IP-Paket an die direkte Gegenstelle geschickt wird, was i.A. nicht vorkommt, da diese Gegenstelle keine Internetdienste anbietet und daher von keinem Client angesprochen wird.

Als endgültigen Schritt, kann man auch das komplette Interface (ippp0) herunterfahren, dann können grundsätzlich keine Verbindungen aufgebaut werden.

9.5 Tips im S.u.S.E. System

Man kann die Runlevel-Scripts natürlich auch manuell benutzen:

```
/sbin/init.d/i4l stop
```

fährt alle ISDN-Netzdevices runter,

```
/sbin/init.d/i4l start
/sbin/init.d/route
```

legt sie wieder an und setzt die Routen.

Wer bei einer syncPPP-Verbindung die Verbindung nur manuell starten möchte, kann eine Eigenschaft des Scriptes `/etc/ppp/ip-up` ab S.u.S.E. 5.2 ausnutzen (Siehe FixMe). Dieses legt beim Verbindungsaufbau eine Defaultroute auf die neu erkannte PtP-Adresse. Beim Verbindungsabbau wird das Device neu angelegt und die Defaultroute gelöscht. Schließlich wird die Datei `/etc/route.conf` durchsucht und die Defaultroute wenn definiert neu angelegt. Definiert man dort keine Defaultroute, so hat man nach Verbindungsabbau eben keine.

Gestartet werden kann dann nur mit dem Kommando:

```
isdnctrl dial ippp0
```

und wer manuell Auflegen will:

```
isdnctrl hangup ippp0
```

9.6 Wie erlaube ich normalen Benutzern Dial-In-Demand zu aktivieren?

Am besten garnicht, denn das ist Aufgabe des Administrators. Es ist nur diesem vorbehalten, Netzdevices und Routen zu konfigurieren.

Versuche nicht, den notwendigen Programmen suid-Attribute zu geben! Erstens ist die Aufgabe sehr schwer, und zweitens handelt man sich damit ein riesiges Sicherheitsloch ein, denn wenn diese Programme erstmal *offen* sind, lassen sich auch andere unerwünschte Dinge damit tun.

Einem einzelnen Script suid-Attribute zu geben, ist unter Linux ebenfalls verboten.

Wer es dennoch unbedingt machen will, der benutze ein Paket wie z.B. `sudo`. Damit lassen sich für einzelne Benutzer bestimmte Kommandos definieren, die diese dann als Benutzer `root` ausführen dürfen.

Hier ein einfaches Beispiel:

1. Paket `sudo` installieren.
2. Mit `visudo` die Konfigurationsdatei editieren, z.B. soll der Benutzer `kfr` das Programm `/usr/local/bin/dial` ausführen dürfen:

```
# User privilege specification
kfr     ALL=/usr/local/bin/dial
```

Hinweis: benutze nur das Kommando `visudo`, um die Konfigurationsdatei (`/etc/sudoers`) zu verändern.

3. Das Script `dial` könnte z.B. so sein:

```
#!/bin/sh

DEVICE=ippp0

if test $UID -ne 0; then
    exec sudo $0 $*
fi

case "$1" in

stop)
    echo stop
    isdnctrl hangup $DEVICE
    ;;
*)
    echo dial
    isdnctrl dial $DEVICE
    ;;

esac
```

   Wird es nicht als User `root` aufgerufen, startet es sich selbst mit `sudo` neu. Mit `dial` wird gewählt, mit `dial stop` wird aufgelegt.

4. sudo fragt beim ersten Start und danach von Zeit zu Zeit das Passwort des aufrufenden Benutzers ab.

5. Um die Passwortabfrage zu verhindern, das Schlüsselwort `NOPASSWD` mit angeben, z.B.

```
kfr     ALL=NOPASSWD:/usr/local/bin/dial
```

10 Konfiguration der Internet-Dienste

Voraussetzung: Die Internet-Verbindung über eine Dial-On-Demand Wählverbindung und das Routing funktioniert.

Jetzt sollen (je nach Bedarf) weitere Internetdienste eingerichtet werden.

10.1 DNS-Cache

Hintergrund: siehe 8 (IP-Nummern Auflösung)

1. Paket `bind` installieren.

2. editiere `/etc/named.boot`:

```
cache . root.cache
options query-log
forwarders 192.76.144.66
```

```
slave
```

Bei `forwarders` werden ein oder mehrere IP-Nummern der Nameserver eingetragen. Die Option `slave` steuert das Verhalten, wenn der Nameserver selbst noch keine Antwort hat, ohne die Option müßte jetzt der eigene Nameserver die Anfrage auflösen (aufwendig). Mit dieser Option (empfohlen) wird dem Forwarder gesagt, daß er soll die Anfrage auflösen. Bei der nächsten Anfrage hat er diese dann im Cache.

Zur Diagnose ist zu empfehlen, noch die Zeile `options query-log` einzufügen, es werden dann über Syslog (also in `/var/log/messages` alle Anfragen an den Nameserver protokolliert, dadurch lassen sich einfach die *Übeltäter* im lokalen Netz finden. Bsp:

```
named[232]: XX /192.168.1.2/www.suse.de/A
```

Der Rechner `192.168.1.2` fragt nach dem A-Record für `www.suse.de`.

3. Wir benutzen uns selbst als Nameserver. Trage als Nameserver die lokale IP-Nummer ein (`192.168.1.1`), siehe 8.3 (Konfiguration der Namensauflösung)

4. Starte den Nameserver:

   - S.u.S.E. Methode: Trage in `/etc/rc.config` ein:

     ```
     START_NAMED=yes
     ```

     Starte Nameserver durch Reboot oder direkt durch `/sbin/init.d/named start`

   - Manuelle Methode: `/usr/sbin/named`

5. Test: `nslookup www.suse.de`.

   Ergebnis: eine Verbindung wird aufgebaut, in messages wird die Anfrage protokolliert und die IP-Nummer wird aufgelöst.

   Eine Wiederholung der Anfrage, wenn die Verbindung nicht besteht, darf keine Verbindung aufbauen, die Anfrage muß sofort beantwortet werden.

10.2 Squid

Squid ist ein WWW- und FTP-Proxy. Der Vorteil eines Proxies liegt nicht nur darin, Anfragen (für mehrere Benutzer) zu cachen, sondern auch darin, daß Clientrechner im lokalen Netz nicht unbedingt echten Internetzugriff (über Masquerading) haben müssen, was die Übersicht und die Sicherheit erhöht.

Squid hat eine Vielzahl von Optionen und Features, die mitgelieferte Beispielkonfiguration in `/etc/squid.conf` ist sehr gut dokumentiert und funktioniert zunächst einmal ohne Änderung.

10.2.1 Starten von Squid

Bei S.u.S.E. wird über die `rc.config`-Variable `START_SQUID` gesteuert, ob Squid gleich beim Systemstart hochgefahren werden soll (über `/sbin/init.d/squid`).

Manuell kann man squid z.B. durch

```
/usr/sbin/squid -sYD >> /var/squid/squid.out 2>&1 &
```

starten.

Vor dem ersten Start muß das Cache-Directory initialisiert werden, dies sollte als Benutzer `squid` geschehen. Als `root` kann man einfach aufrufen:

```
/su squid -c "/usr/sbin/squid -z"/
```

10.2.2 Clients anpassen

Die WWW-Browser müssen konfiguriert werden, damit Sie den Proxy ansprechen. Bei Netscape gibt es die Maske `Options/Network Preferences/Proxies/ Manual Proxy Configuration`. In der Maske gibt man jeweils für FTP und HTTP-Proxy die IP-Nummer des IZG im lokalen Netz ein und als Portnummer `3128` (oder was in `/etc/squid.conf` definiert ist.

Zusätzlich sollte man noch das Feld `No Proxy for` ausfüllen, für welche Domains nicht über den Proxy gegangen, sondern direkt auf den WWW-Server zugegriffen werden soll, z.B.: `localhost isdnworkshop.de`.

10.3 Fetchmail

Das Programm `fetchmail` (Paket `pop`) eignet sich dazu, Mails über das POP3-Protokoll vom Provider abzuholen.

Das Abholen kann auch als normaler User durchgeführt werden, wir holen hier die Mails als Root ab, dadurch läßt sich der Vorgang besser automatisieren. Nach dem Abholen werden die Mails dem lokalen Sendmail übergeben und zugestellt.

Der Mailserver sei mail.provider.de. Es gibt zwei Benutzer asterix und obelix, die auf dem lokalen Rechner eva und maria heissen. Als Passwörter werden (auf dem Mailserver) adam und josef benutzt.

- Lege eine Datei `/root/.fetchmailrc` an:

  ```
  poll mail.provider.de protocol POP3 user asterix password adam is eva
  poll mail.provider.de protocol POP3 user obelix password josef is maria
  ```

- Zum Test starte:

  ```
  fetchmail -v --keep -a
  ```

  Die Option `-v` gibt mehr Ausgaben, die Option `-keep` sorgt dafür, daß die Mails auf dem Server zunächst nicht gelöscht werden.

- Wenn das erfolgreich war, trage in `/etc/ppp/ip-up` das Kommando `fetchmail -a >> /var/log/fetchmail` in der Start-Section ein.

Mehr Infos:

http://www.suse.de/Support/sdb/fetchmail.html

Übung: auf dem Server liegen Mails für jede Workstation bereit. Richte fetchmail so ein, daß bei jedem Verbindungsaufbau Mails abgeholt werden. Prüfe die lokale Zustellung. Siehe `/support-db/sdb/fetchmail.html` und `/etc/ppp/ip-up`.

10.4 Sendmail

Über Sendmail kann man dicke Bücher schreiben (siehe 13.3 (Sendmail)).

Das S.u.S.E. Paket `sendmail` ist für diese Zwecke hier bestens gerüstet. Besonders wichtig sind hier zum einem, daß die Absenderadresse richtig gesetzt wird, denn die lokale Domain könnte ja zur E-Mail-Adresse beim Provider unterschiedlich sein. Zum anderen sollen lokale E-Mails sofort zugestellt werden, Mails die über die Wählleitung verschickt werden müssen, sollen dagegen in eine Queue gestellt werden, ohne daß eine Verbindung aufgebaut wird.

Wie immer gibt es mehrere Wege:

- Sendmail über `/etc/rc.config` konfigurieren:

```
FROM_HEADER="klaus.franken.de"
SENDMAIL_TYPE="yes"
SENDMAIL_SMARTHOST="mail-n.franken.de"
SENDMAIL_LOCALHOST="localhost franken.b.eunet.de glen.home.suse.de \
        klaus.franken.de"
SENDMAIL_RELAY=""
SENDMAIL_ARGS="-bd -om"
SENDMAIL_EXPENSIVE="yes"
SENDMAIL_NOCANONIFY="yes"
```

- Sendmail über m4-Macro-File konfigurieren: Seit sendmail Version 8, bietet Sendmail ein Macro-Paket, bei dem die eigentlich Konfigurationsdatei `/etc/sendmail.cf` nicht *von Hand* erstellt werden muß, sondern über eine Meta-Datei generiert wird. Das Directory ist je nach Distribution unterschiedlich (z.B. `/usr/share/sendmail/m4`, bei S.u.S.E. auch in `/etc/mail`).

  In der Distribution sollten sich Vorlagen befinden. Bei S.u.S.E. ist eine gut kommentierte `/etc/mail/linux.mc` dabei. Bevor man diese ändert, sollte man in `/etc/rc.config` das automatische Generieren abstellen (SENDMAIL_TYPE=no").

  Man generiert eine neue Konfig mit:

```
m4 linux.mc > /etc/sendmail.cf
```

  Mehr Infos: siehe `/etc/mail/README`

- Sendmail Finetuning Bei ausgehenden E-Mails werden abhängig vom lokalen Benutzernamen die E-Mail-Adressen umgeschrieben, Datei `/etc/mail/genericstable`:

```
kfr kfr@klaus.franken.de
sandra sandra@klaus.franken.de
sr sandra@klaus.franken.de
```

  Übung:

  - Schreibe Dir selbst eine Mail auf dem lokalen Rechner
  - Schreibe anderen Usern eine Mail auf dem lokalen Rechner
  - Schreibe eine Mail an `root@server.isdnworkshop.de`
  - Schreibe eine Mail an andere User auf server.isdnworkshop.de (ws0, ws1, ...)
  - Prüfe nach, wo Deine Mails sind

- Stelle sicher, daß Mails beim Verbindungaufbau gequeued verschickt werden, lokale Mails aber sofort zugestellt werden. (Siehe in `/etc/ppp/ip-up`).
- Prüfe die Mailqueue mit `mailq`

10.5 News

Online News lesen ist schon hiermit sehr einfach, als News-Server den Server des ISP angeben. Dazu muß man für die meisten News-Read die Variable `NNTPSERVER` setzen, z.B. `export NNTPSERVER='klaus.franken.de'`. Dies sollte man systemweit in der `/etc/profile` eintragen.

Wünschenswert ist natürlich News-Offline zu lesen und entweder bei Bedarf zu holen bzw. zu verschicken oder dieses per Cron-Job z.B. jede Nacht durchführen zu lassen.

Die Installation eines eigenen News-Servers ist recht aufwendig, es bieten sich **CNews** oder **INN** an. Siehe dazu News HOWTO (fixme).

Ein eigener News-Server ist aber eigentlich nur dann notwendig, wenn man auf diesem selber Newsgruppen einrichten möchte. Will man das nicht, sind CNews und INN vollkommen *overkilled*, deshalb möchte ich hier zwei andere Möglichkleiten vorstellen:

Zwei Pakete bieten sich an: **Leafnode** und **slrn**. Beide sind einfach einzurichten und zu warten und eignen sich für ein mittleres Newsaufkommen vollkommen aus.

slrn ist eigentlich ein eigener News-Reader (textorientiert, sehr flexibel und schnell) und bietet ein eigenes Programm `slrnpull`, das die News abholt und in ein eigenes Spool-Verzeichnis stellt, auf welches direkt von `slrn` zugegriffen werden kann. **Einschränkungen:** es kann kein anderes News-Programm darauf zugreifen; es kann nicht über Netzwerk auf die News zugegriffen werden (vielleicht über NFS, untestet), da kein lokaler News-Server läuft.

Leafnode stellt dagegen einen eigenen News-Server zur Verfügung, braucht aber insgesamt mehr Ressourcen. Der Trick bei **Leafnode** ist der, das sich der Server quasi selbst konfiguriert: wird von einem Client auf eine Gruppe zugegriffen, wird diese automatisch abonniert und ist beim nächsten Abgleich vorhanden; wird dagegen längere Zeit nicht (mehr) auf eine Gruppe zugegriffen, wird diese automatisch gelöscht. Man kann Leafnode also in einem kleineren Netz mit mehreren Lesern trotzdem nahezu unbeaufsichtigt laufen lassen.

Beide Programme arbeiten sehr gut in dieser Dial-On-Demand-Umgebung, Zugriffe auf den News-Server beim Provider werden nur auf Wunsch, nie aber automatisch ausgeführt.

10.5.1 slrn installieren und konfigurieren

Die getestete Version ist 0.9.5.2 von

space.mit.edu:/pub/davis/slrn

Es wird die slang-Bibliothek ab Version 1.0.3 benötigt (bei S.u.S.E. 5.2 ist noch 0.99.38 dabei), zu bekommen unter

space.mit.edu:/pub/davis/slang

Beim Compilieren nicht vergessen auch `make slrnpull` anzugeben. Die Binaries z.B. nach `/usr/local/bin` kopieren, oder folgendes ausführen:

```
install -m 755 -o root -g root src/objs/slrn /usr/local/bin
install -m 755 -o root -g root src/objs/slrnpull /usr/local/bin
install -d /usr/doc/packages/slrn -m 755 -o root -g root
```

```
install -m 644 -o root -g root doc/* /usr/doc/packages/slrn
install -m 644 -o root -g root COPYRIGHT /usr/doc/packages/slrn
install -m 644 -o root -g root COPYING /usr/doc/packages/slrn
install -m 644 -o root -g root README /usr/doc/packages/slrn
install -m 644 -o root -g root changes.txt /usr/doc/packages/slrn
install -m 644 -o root -g root doc/slrn.1 /usr/local/man/man1
install -d /usr/doc/packages/slrn/slrnpull -m 755 -o root -g root
install -m 644 -o root -g root slrnpull/* /usr/doc/packages/slrn/slrnpull
```

Dann das Spool-Verzeichnis anlegen und die Config-Datei erstellen:

```
mkdir /var/spool/slrnpull
cd /var/spool/slrnpull
cp /src/slrn/slrnpull/slrnpull.conf .
```

In `slrnpull.conf` könnte z.B. folgendes stehen:

```
default                0   14
de.alt.comm.isdn4linux
```

Jetzt noch den News-Reader auf diesen Spool-Pfad konfigurieren, in `~/.slrnrc` anfügen (anpassen !):

```
%%% Spool
set spool_inn_root   "/var/spool/slrnpull"
set spool_root       "/var/spool/slrnpull/news"
set spool_nov_root   "/var/spool/slrnpull/news"
set use_slrnpull 1
set read_active 1
set server_object    "spool"
hostname "klaus.franken.de"
set username "kfr"
```

Das Abholen, Verschicken eigener News und das Löschen alter Artikel geschieht mit einem einzigen Kommando (als root), z.B.:

```
slrnpull -d /var/spool/slrnpull -h news.franken.de
```

Beim ersten Mal dauert das natürlich sehr lange und sollte daher manuell ausgeführt werden. Im Betrieb kann man das über einen Croneintrag oder in `/etc/ppp/ip-up` bei jedem Verbindungsaufbau durchführen lassen.

Beim manuellen Start gibt `slrnpull` Meldungen auf der Console aus; wird es im Hintergrund gestartet, loggt es nach `/var/spool/slrnpull/log` (Achtung: diese Datei kann gross werden!).

10.5.2 Leafnode installieren und konfigurieren

Leafnode (Version 1.4) gibt es auf

ftp.troll.no:/pub/freebies/

Die mitgelieferten Dateien `README` und `INSTALL` beschreiben die Installation sehr gut.

Im folgenden Beispiel werden die Binaries `leafnode`, `fetch` und `texpire` nach `/usr/local/bin` installiert (Makefile anpassen!).

Zunächst wird der NNTP-Server `leafnode` in der `/etc/inetd.conf` durch folgende Zeile aktiviert:

```
nntp    stream  tcp     nowait  news    /usr/sbin/tcpd /usr/local/bin/leafnode
```

Danach ein `killall -1 inetd` ausführen.

Als nächstes muß ein User und eine Gruppe `news` angelegt werden, z.B. durch folgenden Eintrag in `/etc/passwd`:

```
news:x:9:13::/var/spool/news:/bin/bash
```

Alle Arbeiten müssen dann als User `news` ausgeführt werden (als Root: `su - news`)!

Im Verzeichnis `/usr/lib/leafnode` wurde bei der Installation eine Bsp-Datei angelegt, die man kopieren und anpassen muss:

```
su - news
cd /usr/lib/leafnode
cp config.example config
```

Die Datei ist kommentiert, hier arbeiten folgende Einträge:

```
server = news.franken.de
expire = 20
maxcount = 1000
```

Jetzt muß man dafür sorgen, daß das Programm `texpire` regelmässig aufgerufen wird (ansonsten werden keine alten News wieder gelöscht), hier arbeitet folgender Crontab-Eintrag vom User root:

```
42  5 * * * su news -c texpire
```

um jede Nacht um 5:42 zu löschen.

Durch das Kommando `fetch` (besser `fetch -v`) wird nun der News-Server initialisiert, aber keine Gruppen sind aktiv.

In dem man jetzt einmalig durch einen News-Reader auf diesen Newsserver und auf die interessanten Gruppen zugreift (es werden natürlich alle mit der Anzahl 0 angezeigt), werden die Gruppen abonniert. Beim nächsten Aufruf von `fetch` werden dann die Artikel geholt.

Auch hier kann man `fetch` via Crontab regelmässig oder durch einen Eintrag in `/etc/ppp/ip-up` aufrufen.

Probleme: man hat keinen direkten Einfluß darauf, welche Gruppen abonniert werden. Es sei denn, daß man vor dem Aufruf von `fetch` das Verzeichnis `/var/spool/news/interesting.groups` *aufräumt*.

Die Ausgabe von fetch sollte beachtet werden, abgelehnte eigene Postings werden nirgens abgespeichert, sondern einfach gelöscht.

10.6 Firewall

Hinweis: Firewalls sind ein heikles Thema. Insbesondere hierfür übernimmt der Autor keine Garantie! Wer eine wirklich sicheres System benötigt, soll zumindest das Firewall HOWTO lesen oder einen Experten dafür beauftragen.

Über Firewalls kann man dicke Bücher schreiben ... (siehe 13.3 (Firewall) oder das *Firewall HOWTO*.

Die einfachste (aber wirkungsvolle) Methode ist die Benutzung eines Paketfilters, die direkt vom Linux-Kernel unterstützt wird und über das Kommando `ipfwadm` (IP-FireWall ADMinistration) konfiguriert wird.

10.6.1 Was ist ein Paketfilter?

Jedes IP-Paket, das vom Kernel behandelt wird, wird nach einer Regelliste untersucht und entweder akzeptiert oder abgelehnt.

Es werden drei verschiedene Listen geführt:

1. Incoming (Schalter `-I`): einkommende Pakete
2. Outgoing (Schalter `-O`): ausgehende Pakete
3. Forwarding (Schalter `-F`): durchgehende Pakete

10.6.2 Wie gibt man eine Firewall-Regel an?

Der `ipfwadm`-Aufruf setzt sich zusammen aus:

- Wann?: Incoming (-I), Outgoing (-O) oder Forwarding (-F)
- Wohin?

  Man kann neue Regeln an den Anfang der Liste (-i) oder an das Ende der Liste (-a). Die Regeln werden immer von vorne nach hinten interpretiert, bei der ersten passenden Regel wird nicht weitergesucht.
- Was tun?

  Soll das Paket akzeptiert werden (accept), oder abgewiesen (deny) werden.
- Protokoll?

  Mögliche Protokolle sind tcp, udp, icmp oder alles (all)
- Quell-IP?

  Angabe des Source-IP-Nummern-Bereiches (-S), z.B. `-S 192.168.42.0/24`
- Ziel-IP?

  Angabe des Ziel-IP-Nummern Bereiches (-D)
- Port?

  Meist wird direkt hinter der Ziel-IP-Nummer noch der Ziel-Port mit angegeben, dies kann der numerische Wert oder der Alias, wie in `/etc/services` definiert.
- Wo?

  Mit dem Schalter -W kann die Regel auf ein Netzdevice beschränkt werden.

Weiterhin gibt es folgende wichtige Optionen:

- -f: zurücksetzen des Reglewerkes für -I, -O oder -F
- -o: beim Zutreffen der Regel wird eine Meldung via syslog in `/var/log/messages` geschrieben.
- -m: Masquerading, s.u.
- -A: Accounting, s.u.
- -l oder -lne: Listet die Regeln.

10.6.3 Was für Regeln brauche ich mindestens?

Eines der größten Sicherheitslöcher ist das sogenannte **Spoofing**. Darunter versteht man, daß ein eigentlich fremder Rechner behauptet eine IP-Nummer aus dem eigenen (sicheren) Netz zu haben. Daher müssen als erstes Regeln definiert werden, die verhindern, daß eigene IP-Nummern aus dem unsicheren Netz hereinkommen können.

Als nächstes sollte man alle Zugriffe von außen verbieten und nur (bei Bedarf) die benötigten Dienste (sendmail, www) freischalten.

10.6.4 Ein einfacher Firewall

Das lokale Ethernet ist auf 192.168.42.0 konfiguriert. Wir erwarten IP-Nummer aus dem Bereich 193.110.3.0/24 zugewiesen zu bekommen, wobei der PtP-Partner nicht aus diesem Bereich ist (sonst würden seine Pakete auch abgewiesen werden)

```
# spoofing verbieten:
/sbin/ipfwadm -I -a deny -o -P all -S 192.168.42.0/24 -D 192.168.42.0/24 -W ippp0
/sbin/ipfwadm -I -a deny -o -P all -S 192.168.42.0/24 -D 193.110.3.0/24 -W ippp0
/sbin/ipfwadm -I -a deny -o -P all -S 193.110.3.0/24 -D 192.168.42.0/24 -W ippp0
/sbin/ipfwadm -I -a deny -o -P all -S 193.110.3.0/24 -D 193.110.3.0/24 -W ippp0


# Zugriffe von ueberall auf den Mail-Server (Port 25) erlauben:
/sbin/ipfwadm -I -a accept -P tcp -S 0/0 -D 192.168.42.1 25 -W ippp0

# Zugriffe von ueberall auf den DNS-Server (Port 53) erlauben:
/sbin/ipfwadm -I -a accept -P tcp -S 0/0 -D 192.168.42.1 53 -W ippp0

# sonst alles verbieten (getrennt fuer Protokoll tcp und udp)
/sbin/ipfwadm -I -a deny -o -P tcp -S 0/0 -D 192.168.42.0/24 1:1023 -W ippp0
/sbin/ipfwadm -I -a deny -o -P tcp -S 0/0 -D 193.110.3.0/24 1:1023 -W ippp0

/sbin/ipfwadm -I -a deny -o -P udp -S 0/0 -D 192.168.42.0/24 1:1023 -W ippp0
/sbin/ipfwadm -I -a deny -o -P udp -S 0/0 -D 193.110.3.0/24 1:1023 -W ippp0
```

Bei S.u.S.E. läßt sich obiges Bsp. auch in der `/etc/rc.config` einstellen:

```
FW_START="yes"
FW_LOCALNETS="192.168.42.0/24 193.110.3.0/24"
FW_MAILSERVER="192.168.42.1"
FW_DNSSERVER="192.168.42.1"
FW_WORLD_DEV="ippp0"
FW_LOG_ACCEPT="no"
```

```
FW_LOG_DENY="yes"
FW_TCP_LOCKED_PORTS="1:1023"
FW_UDP_LOCKED_PORTS="1:1023"
```

Siehe auch `/usr/doc/packages/firewall`

10.7 Masquerading

Masquerading (auch **Network Adress Translation** genannt) braucht man dann, wenn man ein internes Netz mit privaten IP-Nummern hat, vom ISP aber nur eine IP-Nummer (und diese vielleicht sogar dynamisch) bekommt. Die IP-Pakete werden beim rausschicken auf der Internetleitung umgeschrieben und mit der eigenen IP-Nummer versehen. Umgekehrt wird eine Tabelle der offenen Verbindungen gehalten, damit einkommende Pakete wieder dem ursprünglichen Absender zugestellt werden können.

Hat man sich mit dem Firewall (Paketfilter via ipfwadm, s.o.) vertraut gemacht, ist Masquerading fast trivial, denn es findet an derselben Stelle statt und wird fast genauso konfiguriert, es wird lediglich der Schalter `-m` dazugegeben.

Beispiel: Pakete aus dem internen Netzwerk (192.168.42.0/24), die zum Provider (Device `ippp0`) verschickt werden, sollen mit der jeweils gültigen IP-Nummer maskiert werden. Es wird einer Forwarding-Rule der Schalter `-m` mitgegeben:

```
/sbin/ipfwadm -F -a accept -P all -S 192.168.42.0/24 -D 0/0 -m -W ippp0
```

Bei manchen Internet-Diensten (z.B. ftp) wird nicht nur ein Socket geöffnet, sondern auch ein zweiter für die Datenübertragung, die der Server zum Client aufbaut. Da der Client aber selbst nicht erreichbar ist (private IP-Nummer) und der Server die Verbindung zum falschen Rechner (IZG) aufbaut, klappt diese Methode ohne weiteres Wissen über die speziellen Eigenheiten des entsprechenden Protokolls nicht. Abhilfe schaffen dafür spezielle Routinen, die auch dafür *re-maskieren* können. Diese werden durch Kernel-Module geladen:

```
/sbin/insmod ip_masq_cuseeme
/sbin/insmod ip_masq_ftp
/sbin/insmod ip_masq_irc
/sbin/insmod ip_masq_quake
/sbin/insmod ip_masq_raudio
/sbin/insmod ip_masq_vdolive
```

Bei S.u.S.E. läßt sich obiges Bsp. auch in der `/etc/rc.config` einstellen:

```
MSQ_START="yes"
MSQ_NETWORKS="192.168.42.0/24"
MSQ_DEV="ippp0"
MSQ_MODULES="ip_masq_cuseeme ip_masq_ftp ip_masq_irc ip_masq_quake ip_masq_raudio ip_masq_vdolive"
```

Siehe auch `/usr/doc/packages/firewall`

10.8 Accounting

Siehe `man ipfwadm` Stichwort `-A`

10.9 Samba

Samba ist ein File- und Druckerserver für das unter Windows benutzte SMB-Protokoll.

Das Thema gehört also garnicht hier her... doch: denn es kann in unserem Fall Probleme machen.

Beim SMB-Protokoll wird sehr viel mit Broadcasts gearbeitet, die Rechner schicken sich ständig (auch wenn eigentlich keine Aktionen ausgeführt werden) Nachrichten zu. Der Samba-Server wird meist so ausgeliefert, daß dieser alle verwendendbaren Netzdevices benutzt und dorthin Nachrichten schickt, also auch an das ippp0-Device.

Folge: **es werden ständig Verbindungen aufgebaut!**

Abhilfe:

1. Starte Samba nur, wenn Du es auch brauchst. Bei S.u.S.E. wird Samba schon aktiviert, wenn das Paket installiert ist. Setze in `/etc/rc.config`: `START_SMB=no"`
2. Wenn Du es brauchst, sage Samba, welche Devices benutzt werden dürfen.

   In der `/etc/smb.conf` setze z.B, in der `global`-Section: `interfaces = 192.168.2.1/24`

   Mehr Infos:

   http://www.suse.de/Support/sdb/isdn_samba.html

11 Installation

Je nach verwendeter Distribution müssen die Programme und Treiber selbst installiert werden.

11.1 verwendete Programmversionen

- **Kernel**: 2.0.34 - 2.0.36
- **HiSax**: 2.1 (aus 2.0.33/34) bzw. 3.1 (aus 2.0.36)
- **sudo**: 1.5.2

11.2 Unterschiede Kernel 2.0 und 2.2

- Routing: beim 2.0er Kernel wird wie oben ausführlich beschrieben, das Routing für ein Netzdevice gelöscht, sobald eine neue IP-Nummer zugewiesen wurde. Beim 2.2er Kernel ist dies anders gelöst. Hier reicht es, wenn das Routing einmalig beim Booten eingestellt wird, die Dateien `/etc/ppp/ip-up` und `/etc/ppp/ip-down` können daher wesentlich einfacher gehalten werden.
- RST-Provoking-Mode: ist leider nicht im Kernel 2.2 integriert. Siehe: *http://www-users.rwth-aachen.de/Michael.Mueller4/dynip.html*

12 Mailinglisten/News

12.1 Welche Mailinglisten gibt es?

Zwei Mailinglisten beschäftigen sich ausschliesslich mit dem Thema isdn4linux:

- `isdn4linux@hub-wue.franken.de` Dies ist die offizielle Mailing-Liste. Zum subscriben, schicke man eine Mail an

  majordomo@hub-wue.franken.de

  mit `subscribe isdn4linux emailadresse` im **Body** der Mail (Subject ist egal), wobei `emailadresse` die eigene E-Mail-Adresse ist - bitte sorgfältig prüfen.

  Alternativ kann man auch über die Newsgruppe

  de.alt.comm.isdn4linux

  teilnehmen.

  Über den Server *http://www.dejanews.com* kann man diese Mailingliste durchsuchen.

- `suse-isdn@suse.com` Die i4l-Mailingliste speziell für die S.u.S.E.-Distribution.

  Zum subscriben, schicke man eine Mail an *majordomo@suse.com* mit `subscribe suse-isdn emailadresse` im **Body** der Mail (Subject ist egal), wobei `emailadresse` die eigene E-Mail-Adresse ist - bitte sorgfältig prüfen.

  Diese (und weitere S.u.S.E.-) Mailingliste(n) stehen auch über ein WWW-Frontend (zum Lesen) zur Verfügung:

  http://www.suse.com/Mailinglists/index.html

12.2 Wie frage ich auf der Mailingliste?

Je besser man fragt, desto besser ist die Antwort! Schreibe übersichtlich. Niemand liest sich einen ewig langen Text durch, nur um herauszufinden, was überhaupt die Frage ist.

Stelle zunächst sicher, daß die Lösung nicht schon beschrieben ist, es ist unfair und Zeitverschwendung andere zu fragen, wenn man es selbst nachlesen könnte. Siehe 13 (Links) und suche nach einer Lösung.

Auf

http://www.dejanews.com/home_ps.shtml

kannst Du in der Newsgroup `de.alt.comm.isdn4linux` direkt nach einem Stichwort suchen, um zu sehen, ob Dein Problem ev. vor kurzem schon diskutiert und gelöst wurde.

Gib Deine Distribution und die verwendeten Versionsnummern (z.B. Kernel, HiSax) mit an. Gib auch an, was Du schon probiert hast.

Gib exakte Fehlermeldungen, z.B. aus `/var/log/messages`. Niemand kann raten - mit selbst (falsch) interpretierten Meldungen kann niemand etwas anfangen.

12.3 Wie helfe ich auf der Mailingliste?

Möglichst viel und gut :-)

13 Links

13.1 WWW und FTP

- Homepage vom *ISDN-HOWTO*: *http://www.franken.de/users/klaus/DE-ISDN-HOWTO/html/DE-ISDN-HOWTO.html*

- die englische Version (noch in Arbeit) des *ISDN-HOWTO*: *http://www.nordkom.netsurf.de/Scott.Hanson/EN-i4l.h*
- Deutsches Linux HOWTO Projekt *http://www.tu-harburg.de/~semb2204/dlhp/*
- Die S.u.S.E. Support-Datenbank *http://www.suse.de/Support/sdb*
- Hinweise zu fetchmail: *http://www.suse.de/Support/sdb/fetchmail.html*
- Hinweise zu ungewollten Verbindungsaufbauten *http://www.suse.de/Support/sdb/isdn_dial.html*
- Bernd-Hailers *Leafsite* Dokumentation: *http://www.lrz-muenchen.de/~ui161ab/www/isdn/*
- Weitere Beispiel-Konfigurationen: *http://www.rosat.mpe-garching.mpg.de/~web/ISDN.html*
- RST-Provoking Patch: *http://www.image.dk/~ehcorry/linux/*, siehe auch 6.1 (Der RST-provoking mode).
- Michael Hipp's ISDN-Seite (ipppd) *http://www.sfs.nphil.uni-tuebingen.de/~hipp/isdn/isdn.html*
- Der offizielle FTP-Server: *ftp.franken.de:/pub/isdn4linux*
- Die aktuellsten Version der Utils und vom HiSax: *ftp.suse.com:/pub/isdn4linux* (Siehe auch *ftp.suse.com:/pub/isdn4linux/README*!) Hier wird der CVS-Tree (Entwicklerbaum) tagesaktuell als tgz-File eingepackt.
- Das S.u.S.E. i4l-Paket und die S.u.S.E.-Scripts: *ftp.suse.com:/pub/SuSE-Linux/5.2/suse/n1* (Statt `5.2` jeweils die aktuelle Version einsetzen.) Interessant sind die Paket `i4l.rpm`, Basispaket: *ftp.suse.com:/pub/SuSE-Linux/5.2/suse/n1/i4l.rpm* `i4ldoc.rpm`, Documentation: *ftp.suse.com:/pub/SuSE-Linux/5.2/suse/doc1/i4ldoc.rpm* `i4lfirm.rpm`, Firmware für aktive Karten: *ftp.suse.com:/pub/SuSE-Linux/5.2/suse/n1/i4l.rpm*
- AVM-B1 FTP-Server: *calle.in-berlin.de:/pub/linux/isdn*
- *ISDN FAQ*: *http://www.suse.de/doku/i4l-faq/index.html*
- kISDN: *http://www.physik.uni-bielefeld.de/~twesthei/kISDN.htm*
- *ISA PnP HOWTO*: *http://www.suse.de/Support/sdb/rb_isapnp.html*
- Telefonanrufe als WinPopUp-Fenster (Win95/NT) melden: *http://linux0.urz.uni-heidelberg.de/~mseuffer/data*
- ISDN-Kabel selber machen: *http://uriela.in-berlin.de/~hifi/faqkabel.html#isdn*
- Zeit synchronisieren (Chrony) *http://www.curnow.demon.co.uk/chrony/index.html*

13.2 lokale Dokumentationen

Siehe (bei S.u.S.E.) `/usr/doc/packages/i4l` und `/usr/doc/packages/i4ldoc` (die FAQ, Paket `i4ldoc`).

Hilfesystem (Start mit `hilfe`), insbesondere die Support-DB unter der URL `http://localhost/support-db/sdb`

Sind die Kernelquellen installiert, steht im Verzeichnis `/usr/src/linux/Documentation/isdn` sehr viel nützliches.

13.3 Bücher

- **Sendmail**: (Fledermausbuch), O'Reilly
- **Firewall**: (), O'Reilly
- **Technik der Netze**: Gerd Siegmund, 3. Aufl.

14 Credits

An diesem HOWTO haben mitgewirkt:

- ***Carsten Schwertfeger***: Korrekturen
- ***Rainer Gievers***: Hinweis auf **Technik der Netze** von Gerd Siegmund

Bedanken möchte ich ich mich bei:

- ***Karsten Keil*** für seinen unermüdlichen Einsatz beim HiSax-Treiber
- ***Fritz Elfert*** für ISDN4linux
- ***Michael Hipp*** für den ipppd
- ***Erik Corry*** für den RST-Provoking-Patch
- der **Mailingliste und Newsgruppe** isdn4linux@hub-wue.franken.de
- bei **S.u.S.E.**
- und vielen vielen weiteren Entwicklern.

15 News

Änderungen im HOWTO:

15.1 v1.2, 02. September 1998

- Typos von Carsten Schwertfeger integriert
- Hinweis auf Kernel 2.1, Routing
- einige neue Links

15.2 v1.3, 18. Januar 1999

- Hinweis auf Unstimmigkeit bei TK-Anlagen von Lorenz Hahn

15.3 v1.4, 1. März 1999

- Hinweise auf Anführungszeichen im PPP-Passwort von Joerg Scherer.

15.4 v1.5, 26. Mai 1999

- Umstellung auf CHAP (anstatt PAP).