Linux PPP HOWTO

Robert Hart (`hartr@hedland.edu.au`) und Marco Budde (`Budde@tu-harburg.de`) v1.0-2, 3. Januar 1997

Dieses Dokument beschreibt, wie man einen Linux PC mit einem PPP Server verbindet.

Inhaltsverzeichnis

1 Einleitung

1.1 Neue Versionen dieses Dokuments

Die jeweils neuste Version dieser HOWTO ist auf dem WWW Server des Deutschen Linux HOWTO Projekts unter folgender Adresse zu finden:

`http://www.tu-harburg.de/dlhp/`

1.2 Feedback

Wenn Sie irgendwelche Frage oder Kommentare zu diesem Dokument haben, erreichen Sie mich unter den e-mail Adressen

Internet: `Budde@tu-harburg.de`
Fido: Marco Budde `2:240/6298.5`

Ich freue mich über alle Vorschläge und Kritiken zu dieser HOWTO. Wenn Ihnen also irgendein Abschnitt in dieser HOWTO unklar sein sollte, schreiben Sie mir bitte.

1.3 Copyright

Dieses Dokument ist urheberrechtlich geschützt. Das Copyright für die englische *PPP HOWTO*, auf der dieses Dokument basiert, liegt bei Robert Hart. Das Copyright für die deutsche Übersetzung liegt bei Marco Budde.

Die englische HOWTO darf nach folgender Lizenz verbreitet werden:

> The copyright of this document is retained by the author. Permission is granted to distribute the document by electronic means and on CDs provided that it is kept entirely in its original format. Permission is also granted to print a copy of this document for personal use.
>
> The republishing of this document in part or in whole without the permission of the copyright holder by any means other than as noted above is prohibited.

Der Autor der deutschen Übersetzung erlaubt die Verbreitung seiner Übersetzungsarbeit an dieser HOWTO unter der GNU *General Public License*.

2 Grundbegriffe

PPP (Point to Point Protocol) ist ein Protokoll, um das Internet Protokoll (IP) und andere Netzwerk Protokolle über eine serielle Verbindung benutzen zu können. Dabei kann es sich bei der seriellen Verbindung um eine direkte Verbindung zwischen zwei Rechner oder eine Verbindung mittels Modems handeln. Mit PPP kann man den eigenen Linux PC mit einem PPP Server verbinden und auf die Resourcen des Netzwerkes zugreifen, mit dem der Server verbunden ist, als ob man direkt einen Rechner in diesem Netzwerk benutzen würde.

Einer der größten Unterschiede ist natürlich die Geschwindigkeit. Eine Ethernetverbindung arbeitet mit maximal 100 MBit pro Sekunde, während ein Modem mit Geschwindigkeiten bis zu 33,6 KBit pro Sekunde arbeitet. In Abhängigkeit vom Typ der PPP Verbindung gibt es eventuell einige Einschränkungen in der Benutzung von einigen Applikationen oder Servicen.

2.1 Klients und Server

Wenn man sich in einen Rechner einwählt, um eine PPP Verbindung aufzubauen, dann ist man ein Klient. Der Rechner, mit dem man eine Verbindung aufbaut, wird Server genannt.

Wenn man selbst einen Linux Rechner so konfiguriert, daß er einem anderen Benutzer ermöglicht, sich in das Netzwerk per Modem einzuwählen, dann installiert man einen PPP Server.

Jeder Linux PC kann sowohl PPP Server als auch Klient sein. Dabei kann er beides gleichzeitig sein, wenn man über mehr als eine serielle Schnittstelle (und wenn notwendig Modems) verfügt. Für das PPP Protokoll selbst gibt es eigentlich keinen wirklichen Unterschied zwischen einem Klient und einem Server, wenn die Verbindung erst einmal aufgebaut ist. Das bedeutet, daß PPP dafür benutzt werden kann, zwei oder mehrere lokale Netzwerke (LANs) untereinander zu verbinden oder ein oder mehrere LANs mit dem Internet zu verbinden.

Im Interesse der Verständlichkeit wird in diesem Dokument der Rechner, der die Verbindung einleitet, sich also einwählt, als *Klient*, und der Rechner, der den Anruf beantwortet und die Zugangsberechtigung des Anrufers unter Verwendungen der Benutzerkennung, Paßwörtern und eventuell anderer Methoden überprüft, als *Server* bezeichnet.

Die meisten Leute interessieren sich wahrscheinlich dafür, wie man mit PPP als Klient einen oder mehrere Rechner mit dem Internet verbindet. Die Prozedur, die in diesem Dokument beschrieben wird, erlaubt es, eine Internet Verbindung einzurichten und zu automatisieren.

2.2 IP Nummern

Jedes Netzwerkinterface, das mit dem Internet verbunden ist, muß seine eigene, einzigartige IP Nummer haben. Für jedes Land gibt es eine Organisation, die diese IP Nummern zentral vergibt.

Bitte beachten Sie, daß die in diesem Dokument benutzen IP Nummern alle *fiktiv* sind! Wenn man eine Verbindung zum Internet aufbauen möchte, muß man eine offizielle IP Nummer verwenden, die einem vom Provider zugewiesen worden ist. Man darf sich auf keinen Fall einfach irgendeine IP Nummer ausdenken und diese verwenden, da dieses zu erheblichen Störungen führt.

Es gibt IP Nummern, die speziell für LANs gedacht sind, die nicht mit dem Internet verbunden sind. Eine dieser Folgen ist 10.0.0.0 bis 10.255.255.255. Dieses ist auch die Folge, die in den Beispielen in diesem Dokument verwendet wird.

Für die meisten Anwender, die einen einzelnen Rechner oder ein kleines LAN über PPP mit einem Internet Service Provider (ISP) verbinden, ist es nicht notwendig, eine eigene IP Nummer (oder genauer eine Netzwerk Nummer) zu beantragen.

Wenn man ein kleines LAN mit dem Internet verbinden möchte, können viele Internet Service Provider ein entsprechendes Subnet (eine spezielle Folge von IP Nummern) von ihrem existierenden IP Adressraum zur Verfügung stellen.

Für Anwender, die einen einzelnen PC über einen Internet Service Provider mit dem Internet verbinden möchten, benutzen die meisten Provider eine dynamische IP Nummern Vergabe. Als Teil des Verbindungsaufbaues nennt der PPP Servern nun dem eigenen Rechner die IP Adresse, die er für die aktuelle Verbindung benutzen soll. Man erhält also nicht bei jeder Verbindung mit dem Server die gleiche IP Nummer. Dieses hat Konsequenzen für Programme auf dem eigenen Rechner, die als Server fungieren, wie `sendmail`, `ftpd`, `httpd` und so weiter. Die Beschränkungen von Servicen aufgrund von einer dynamische IP Nummern Vergabe und Wege diese, wenn möglich, zu umgehen, werden wir in einem späteren Teil dieses Dokument erläutern.

3 Ziel dieses Dokuments

Dieses Dokument versucht die Anwender zu unterstützen, die Linux und PPP benutzen möchten, um sich in einen PPP Server einzuwählen und eine IP Verbindung mittels PPP aufzubauen. Es wird davon ausgegangen, daß der Anwender

PPP bereits kompiliert und auf seinem Linux Rechner installiert hat. Allerdings werden wir trotzdem darauf eingehen, wie man das Kernel neu konfiguriert und kompilieren muß, um die PPP Unterstützung zu aktivieren.

Obwohl auch DIP, die Standardmethode um eine SLIP Verbindung aufzubauen, benutzt werden kann, um eine PPP Verbindung aufzubauen, sind DIP Skripte meistens sehr kompliziert. Aus diesem Grund geht dieses Dokument *nicht* auf die Verwendung von DIP zum Aufbau einer PPP Verbindung ein.

Stattdessen beschreibt dieses Dokument die Standard Linux PPP Software (`chat`/`pppd`).

Zur Zeit geht dieses Dokument *nicht* auf folgende Themen ein:

- Anschluß und Konfigurationen eines Modems unter Linux (im Detail). Informationen hierüber sind in der *Serial HOWTO* zu finden.
- Installation eines PPP Servers unter Linux. Eine englische *PPP Server HOWTO* befindet sich in der Entwicklung und wird bei Erscheinen diesem Dokument hinzugefügt werden.
- Die Verwendung von DIP um PPP Verbindung aufzubauen. Stattdessen sollte `chat` verwendet werden.
- Benutzung von PAP und CHAP zur Authentifizierung von PPP Verbindungen. Dieses wird zu einem späteren Zeitpunkt hinzugefügt werden.
- Benutzung von socks oder IP Masquerade. Es existieren bereits sehr gute Dokumente, die diese beiden Pakete beschreiben.

4 Beschriebene Software Versionen

Diese HOWTO geht davon aus, daß der Benutzer ein 1.2.x Kernel mit der PPP 2.1.2 Software verwendet. Zur Zeit befindet sich Linux 1.3.x noch in der Entwicklung und der Autor muß selbst mit diesem noch experimentieren.

Es ist auch bereits die PPP Version 2.2 verfügbar. Obwohl diese mit Linux 1.2.x verwendet werden kann, wurde sie für Linux 1.3.x entwickelt, so daß das Kernel für die Verwendung mit Linux 1.2.x gepatcht werden muß.

Dieses Dokument ist speziell für die Unterstützung von neuen Benutzern entwickelt worden und es wird dringend empfohlen, daß der Anwender eine Linux Version und eine passende PPP Version verwendet, die beide stabil zusammenarbeiten.

Sobald der Autor die Zeit findet, mit Linux 1.3.x und PPP 2.2 zu experimentieren, werde zusätzliche Informationen zu diesen diesem Dokument hinzugefügt. Aber bitte fragen Sie den Autoren nicht nach diesen Informationen, denn er wird sie veröffentlichen, wenn sie fertig sind - solche Anfragen werde nach `/dev/null` weitergeleitet.

Falls Sie in der Zwischenzeit Linux 1.3.x benutzen müssen, lesen sie bitte entsprechenden Dokumente. Die hier gegebenen Informationen zur Kernel Konfiguration sind für die 1.3.x Kernel vollkommen falsch.

5 Andere nützliche und wichtige Dokumente

Den Benutzern wird empfohlen, folgendes zu lesen:

- die Dokumentation, die mit dem PPP Paket ausgeliefert wird
- die pppd und chat man pages
- das Linux *Network Administration Guide* (NAG)
- die *NET-3 HOWTO*.

Man kann natürlich dieses Dokument benutzen, um eine PPP Verbindung zu erzeugen, ohne eines der obigen Dokumente gelesen zu haben, allerdings verbessern diese das eigene Verständnis dessen, was abläuft. So hat man die Möglichkeit, viele Probleme selber zu lösen oder wenigstens detailiertere Fragen in den `de.comp.os.unix.linux` Newsgruppen zu stellen.

Diese Dokumente (genauso wie die vielen anderen, inklusive der relevanten RFCs) bieten zusätzliche und detailiertere Erklärungen wie in dieser HOWTO möglich sind.

Falls Sie ein LAN unter der Verwendung von PPP an das Internet anbinden möchten, benötigen Sie ein gutes Fachwissen über TCP/IP Netzwerke. Zusätzlich zu den obigen Dokumenten können die O'Reilly Bücher *TCP/IP Netzwerk Administration* und *Einrichten von Internet Firewalls* sehr hilfreich sein.

6 Konfiguration des Kernel

Um PPP benutzen zu können, muß das Linux Kernel mit PPP Unterstützung neu kompiliert werden. Als erstes muß der Linux Kernel Source Code auf dem Rechner installiert werden, falls dieser nicht bereits installiert ist. Gemäß dem Standard sollte sich der Source Code im dem Verzeichnis `/usr/src/linux` befinden.

In diesem Verzeichnis findet man mindestens eine `README` Datei. Diese enthält eine sehr gute Erklärungen, wie man das Kernel konfiguriert und kompiliert. Diese Datei sollte man auf jeden Fall lesen. Es ist eventuell eine gute Idee, diese Datei auszudrucken und zur Hand zu haben, wenn man ein neues Kernel kompiliert und dieses noch nicht so oft getan hat, um die notwendigen Schritte zu kennen.

Ein kleiner Hinweis - der aktuelle (Linux 1.2.13)

```
make config
```

Dialog erlaubt es *nicht*, rückwärts zu gehen. Falls man also einen Fehler bei der Beantwortung einer der Fragen gemacht hat, muß man der Konfiguration mit `STRG-C` abbrechen und neu starten.

Man *muß* für das Kompilieren des Kernels wissen, was für Karten/Devices sind in dem Rechner befinden.

Um PPP benutzen zu können, muß das Kernel mit PPP Unterstützung konfiguriert werden. PPP erfordert sowohl den `pppd` als auch Kernel Unterstützung für PPP.

Das folgende ist ein Einzug von dem Linux 1.2.13 `make config`:

```
*
* Network device support
*
Network device support? (CONFIG_NETDEVICES) [n] y
Dummy net driver support (CONFIG_DUMMY) [n] y
SLIP (serial line) support (CONFIG_SLIP) [n] n
PPP (point-to-point) support (CONFIG_PPP) [n] y
```

Falls man sich um die Sicherheit seines Computers und/oder LANs sorgt, während man mit dem Internet verbunden ist, möchten man vielleicht Unterstützung für IP Firewalls hinzufügen.

Die anderen `make config` Fragen beantwortet man entsprechend der verwendeten Hardware in dem Rechner und der Features von Linux, die man verwenden möchten. Danach kann man den Anweisung in der `README` folgen, um das Kernel zu kompilieren und zu installieren.

Falls der Linux Rechner als PPP Server oder Internet Gateway eingesetzt werden soll, sollte man IP forwarding einbinden (dieses wird benötigt!).

Außerdem hat das aktuelle Kernel nur 4 PPP Devices. Wenn mehr Devices benötigt werden, muß man den Kernel PPP Source Code verändern. Weitere Informationen hierzu findet man in der `README.linux` Datei, die als Teil der PPP-2.1.2 Distribution ausgeliefert wird.

Wenn erst einmal das neue Kernel installiert und der Rechner rebootet ist, kann man mit der Konfiguration und dem Testen der PPP Verbindung beginnen.

7 Vom Provider zu erfragende Informationen

Bevor man eine PPP Verbindung zum Server aufbauen kann, muß man folgende Informationen vom Systemadministrator oder dem Benutzersupport des Providers in Erfahrung bringen:

- Die Telefon Nummer, die man zur Einwahl benutzen muß. Wobei die meisten Provider getrennte Nummern für analoge Modems und ISDN verwenden.
- Benutzt der Server *dynamische* oder *statische* IP Nummer? Wenn der Server statische IP Nummern verwendet, muß man in Erfahrung bringen, welche IP Nummern man für die Verbindung benutzen muß.
  Die meisten Internet Service Provider benutzen dynamische IP Nummern. Wie oben erwähnt hat das einige Begleiterscheinung in bezug auf die Services, die man benutzen kann.
- Wenn man eine statische IP Nummern verwendet, benötigt man auch eine Netzwerkmaske (network mask).
- Welches sind die IP Nummern der Domain Name Server (DNS) des Providers? Es sollte mindestens zwei geben.
- Benötigt der Server die Verwendung von PAP/CHAP?
- Startet der Server automatisch PPP oder muß man irgendwelche Befehle eingeben, um PPP auf dem Server zu starten, wenn man sich erst einmal eingeloggt hat? Wenn man einen Befehl zum Starten von PPP eingeben muß, wie lautet dieser?

Diese Informationen sollte man sich sorgfältig aufschreiben, da man sie später benötigt.

7.1 Testen der Modem Verbindung

Man sollte sich versichern, daß das Modem korrekt eingerichtet ist und einem bekannt ist, an welcher seriellen Schnittstelle das Modem angeschlossen ist.

- DOS COM1 = Linux `/dev/cua0` (und `/dev/ttyS0`)
- DOS COM2 = Linux `/dev/cua1` (und `/dev/ttyS1`)
  usw.

Man benutzt nun als erstes ein Terminal Programm wie `minicom` und wählt sich in den PPP Servern ein, mit dem man später eine PPP Verbindung aufbauen möchte. Zu diesem Zeitpunkt versuchen wir *nicht* eine PPP Verbindung aufzubauen. Stattdessen wollen wir nur testen, ob wir die richtige Telefonnummer besitzen. Außerdem interessieren wir uns dafür, was der Server an uns sendet, wenn wir uns Einzuloggen und PPP starten.

Während dieses Prozesses speichert man entweder den gesamten Login Prozeß in einer Datei mit oder man schreibt sehr sorgfältig mit, was der Remote Server genau ausgibt, um dem Benutzer mitzuteilen, daß jetzt die Zeit ist, den Benutzernamen, das Paßwort und die Befehle zum Verbindungsaufbau einzugeben.

Es ist sinnvoll, sich mindestens zweimal einzuwählen, da einige Server ihre Ausgaben jedesmal ändern, wenn man sich einloggt. Es gibt zwei wichtige Ausgaben, die der Linux Rechner des Benutzer jedesmal beim Einloggen erkennen können muß:

- Die Ausgabe, die den Benutzer auffordert, seinen Benutzernamen einzugeben.
- Die Ausgabe, die den Benutzer auffordert, sein Paßwort einzugeben.

Wenn der Server automatisch das PPP Protokoll startet, wenn man sich eingeloggt hat, dann sieht man einige wirre Zeichen auf dem Bildschirm. Dieses ist der PPP Server, der dem eigenen Rechner Informationen schickt, um die PPP Verbindung aufzubauen.

Dieses sollte ungefähr so aussehen:

```
~y}#.!}!}!} }8}!}$}%U}"}&} } } } }%}& ...}'}"}(}"} .~~y}
```

An diesem Punkt kann man das Modem auflegen lassen (gewöhnlich, in dem man schnell +++ eingibt und das `ATHO` Kommando eingibt, sobald das Modem mit `OK` antwortet).

Bei einigen Systemen muß PPP expliziert auf dem Server gestartet werden. Dieses liegt darin begründet, daß der Server so installiert worden ist, daß er Benutzer zum einen einen Shellaccount hat und zum anderen das PPP Protokoll verwenden kann. Falls dieses der Fall ist, gibt man den Befehl ein, sobald man eingeloggt ist. Noch einmal, wenn man solchen Müll auf dem Bildschirm sieht, ist das ein Zeichen dafür, daß der PPP Server beim Provider gestartet wurde, so daß man nun auflegen kann.

Falls man diesen Müll *nicht* auf dem Bildschirm sieht, wenn der Server PPP startet, ist es ziemlich sicher, daß man irgendetwas falsch gemacht hat.

Allerdings sind einige PPP Server so eingerichtet, daß sie sich passiv verhalten und nichts senden, bis der Klient, der eigene Computer, den PPP Prozeß startet.

Trotzdem, die Mehrzahl der Server ist aktiv und man sollte den Müll sehen.

Falls man sein Modem nicht zum korrekten Funktionieren überzeugen kann, sollte man das Handbuch des Modems, die man pages für die Kommunikationssoftware und die *Serial HOWTO* lesen. Wenn man das Problem erst einmal gelöst hat, fährt man, wie oben beschrieben, fort.

8 Authentifizierung mit PAP and CHAP

Es ist möglich, einen PPP Server so einzurichten, daß die Benutzer Identifizierung vom PPP Protokoll selbst unter Verwendung von PAP/CHAP übernommen wird. Dabei werden die Identifizierungs Informationen zwischen den beiden PPP Daemonen ausgetauscht.

Dieser Abschnitt muß noch hinzugefügt werden.

9 Anmerkung über serielle Schnittstellen und Geschwindigkeiten

Wenn man ein schnelles Modem mit 14.400 bps oder mehr benutzt, benötigt man eine serielle Schnittstelle, die die anfallende Datenmenge, die ein solchen Modems produzieren kann, transportieren kann. Aus diesem Grund muß die serielle Schnittstelle einen modernen UART (Universal Asynchronous Receiver Transmitter) wie den 16550(A) benutzen. Wenn man einen alten Rechner oder eine alte serielle Schnittstellenkarte verwendet, ist es durchaus möglich, daß die serielle Schnittstelle nur über einen 8250 UART verfügt. Dieses wird wahrscheinlich zu Problemen bei der Benutzung des Modems führen.

Der Befehl

```
setserial -a /dev/ttySx
```

kann benutzt werden, damit Linux den verwendeten UART Typ ausgibt. Falls es sich nicht um einen UART vom 16550 Typ handelt, sollte man diesen auf jeden Fall ersetzen. Wenn es sich bei dem alten UART um einen 16450 handelt und dieser gesockelt ist, kann man ihn einfach durch einen 16550 IC ersetzen. Ansonsten sollte man sich eine neue Multi-IO Karte zulegen, die man für unter 50 DM bekommen kann.

10 Konfiguration des Modems

Man muß das Modem korrekt für PPP konfigurieren. Um dieses zu tun, sollte man das Handbuch des Modems lesen. Die meisten Modems werden mit einer Standardeinstellung ausgeliefert, das bereits die für PPP benötigten Optionen richtig konfiguriert. Die wichtigste Einstellung ist:

- Hardware flow control (RTS/CTS) (`&K3` bei vielen Hayes Modems)

Andere Einstellung (in Standard Hayes Befehlen), die man genauer untersuchen sollte:

- `E1` Command Echo AN (wird von `chat` benötigt)
- `Q0` Report result codes (wird von `chat` benötigt)
- `S0=0` Auto Answer AUS
- `&C1` Carrier Detect AN nur nach Connect
- `&S0` Data Set Ready (DSR) immer AN
- (verschieden) Data Terminal Ready

Um die Konfiguration des Modems zu ermitteln und zu verändern, verwendet man ein beliebiges Terminalprogramm wie `minicom`. Bei vielen Modems kann man die aktuelle Konfiguration durch Eingabe von AT`&V` ermitteln. Sollte dieses nicht funktionieren, hilft das Handbuch des Modems sicherlich weiter. Falls man das Modem völlig verkonfiguriert hat, kann man meistens durch die Eingabe des Befehls AT`&F` die sogenannte *factory settings* des Modems laden. Dieses ist die Konfiguration, mit der das Modem ausgeliefert wurde. Bei den meisten Modems wird hierdurch genau die Konfiguration geladen, die man für PPP benötigt. Allerdings sollte man dieses immer überprüfen.

Nachdem man alle Paramter richtig eingestellt hat, muß man die Konfiguration im nicht flüchtigen Speicher des Modems abspeichern, so daß die Einstellungen auch durch das Ausschalten des Modems nicht verloren gehen. Dieses geschieht bei den meisten Modems durch die Eingabe des AT`&W` Befehls.

Wenn die richtig Modem Konfiguration in diesem abgespeichert worden ist, wird diese bei jedem Reset des Modems aktiviert. Auf diese Weise vereinfacht sich das benötigte `chat` Skript.

10.1 Anmerkungen zur Serial Flow Control

Wenn die Daten über die serielle Verbindung reisen, kann es passieren, daß die Daten schneller ankommen, wie der Computer sie verarbeiten kann. Der Computer ist vielleicht mit etwas anderem beschäftigt; man sollte daran denken, das Linux ein Multitasking und -user Betriebssystem ist. Um sicherzustellen, daß keine Daten verloren gehen, ist eine Methode zur Kontrolle des Datenflußes notwendig.

Es gibt zwei verschiedene Methoden, dieses bei einer seriellen Verbindung zu erreichen:

- Hardware Signale (Clear To Send/Request to Send - CTS/RTS)
- Software Signale (control S und control Q).

Während das zweite vielleicht für eine Terminal (Text) Verbindung geeignet sein mag, werden bei einer PPP Verbindung alle 8 Bits der Leitung verwendet. Und es ist sehr wahrscheinlich, daß sich irgendwo in den übertragenen Daten sich die Bytes der Steuerzeichen control S und control Q befinden, so daß, wenn das Modem die *software flow control* verwendet, es zu Problemen kommen kann.

Für PPP, welches mit 8 Bit Daten arbeitet, ist die *hardware flow control* lebensnotwendig.

11 PPP Optionen Datei

Die Informationen, die benutzt werden, wenn PPP gestartet wird, können auf der Kommandozeile angegeben werden oder in einer Datei gespeichert werden, die der PPP Daemon beim Start einliest.

11.1 Benutzung von PPP und root Rechte

Weil der PPP Daemon Netzwerk devices einrichten muß, die Routingtabelle des Kernel verändern muß und so weiter, benötigt es `root` Rechte, um dieses zu tun.

Das `pppd` Programm sollte mit setuid root installiert werden:

```
-r-sr-xr-x   1 root     root        95225 Jul 11 00:27 /usr/sbin/pppd
```

Falls `/usr/sbin/pppd` nicht mit diesen Rechten installiert ist, sollte man als `root` folgendes Kommando eingeben:

```
chmod u+s /usr/sbin/pppd
```

Wenn es jedem Benutzer möglich sein soll, eine PPP Verbindung aufzubauen, sollte man die `ppp-on/off` Skripte world read/execute machen. Dieses ist jedoch eigentlich nur auf Einzelplatzsystemen sinnvoll.

Wenn man allerdings *nicht* möchte, daß jeder Benutzer die Möglichkeit hat, eine PPP Verbindung aufzubauen, was z.B. sinnvoll sein kann, wenn man nicht möchte, daß die eigenen Kinder sich ohne Kontrolle ins Internet einwählen können, muß man eine PPP Gruppe einrichten, indem man `/etc/group` editiert und folgendes tun:

- Die PPP Gruppe zum Besitzer der `ppp-on/off` Skripte machen.
- Die `ppp-on/off` Skripte lesbar und ausführbar für die PPP Gruppe machen.
- Alle anderen Zugriffsrechte auf die `ppp-on/off` Skripte löschen.
- Die Benutzer, die eine PPP Verbindung aufbauen dürfen, nehme man in die PPP Gruppe von `/etc/group` auf.

```
-rwxr-x---   1 root     PPP          587 Mar 14  1995 /usr/sbin/ppp-on
-rwxr-x---   1 root     PPP          631 Mar 14  1995 /usr/sbin/ppp-off
```

11.2 Konfiguration der Dateien

Um die für PPP benötigten Verzeichnisse zu erzeugen und die notwendigen Datei zu editieren, muß man sich als `root` einloggen. Dieses ist auch notwendig, wenn PPP für alle Benutzer verfügbar sein soll.

Im Verzeichnis `/etc` sollte sich folgendes Verzeichnis befinden:

```
drwxrwxr-x   2 root     root         1024 Oct  9 11:01 ppp
```

Falls es noch nicht existiert, muß es erzeugt werden.

Wenn das Verzeichnis bereits besteht, sollte es eine Template Optionen Datei enthalten, die `options.tpl` heißt. Die Datei befindet sich am Ende dieses Dokumentes. Drucken Sie diese aus, da sie eine Beschreibung alle PPP Optionen enthält. Es ist sinnvoll diese zusammen mit der pppd man page zu lesen. Während man diese Datei als Basis für die eigene `/etc/ppp/options` Datei verwenden kann, ist es wahrscheinlich meistens besser, eine eigene Optionen Datei zu erzeugen, die nicht alle Kommentare des Templates enthält, da die Datei so erheblich kürzer ausfallen wird und leichter zu lesen und zu benutzen ist.

Wenn an den Rechner mehrere serielle Schnittstelle und Modems angeschlossen sind, was typischerweise bei PPP Servern der Fall ist, erzeugt man eine leere `/etc/ppp/options` Datei und jeweils eine individuelle Optionen Datei für jede serielle Line, über die eine PPP Verbindung aufgebaut werden soll. Die Dateien haben die Namen `options.ttyS1`, `options.ttyS2` und so weiter. Wenn es für alle Lines bestimmt gemeinsame Optionen gibt, dann können diese in der `options` Datei und die Port spezifischen Optionen in den entsprechenden `options.ttySx` Dateien abgelegt werden.

Für eine einzelne PPP Verbindung kann man aber einfach nur die `/etc/ppp/options` Datei verwenden. Alternativ kann man dem `pppd` alle Informationen auch auf der Kommandozeile übergeben.

Es ist einfacher ein Setup zu verwalten, das die `/etc/ppp/options.ttySx` Dateien verwendet. Wenn man PPP Verbindungen zu mehreren verschiedenen sites aufbaut, kann man für jede site eine eigene Optionen Datei als `/etc/ppp/options.site` erzeugen und diese Optionen Datei als Parameter dem PPP Befehl beim Verbindungsaufbau übergeben.

11.3 Welche Optionen sollte ich benutzen?

Wie bei allen Dinge hängt das vom Anwendungszweck ab. Das folgende ist eine einfache Standard Optionen Datei, die für die meisten Klient Verbindungen, die nicht PAP/CHAP benutzen, funktionieren sollte.

Falls sie trotzdem nicht arbeiten sollte, lesen Sie bitte die Template Datei und die pppd man pages und fragen sie beim Systemadministrator oder Support nach, die den Server betreiben, in den Sie sich einwählen möchten.

```
# /etc/ppp/options (kein PAP/CHAP)
#
# benutze die modem control lines
modem
# benutze hardware flow control
crtscts
# erzeuge eine default route fuer die Verbindung in der Routingtabelle
defaultroute
# alle Zeichen der Verbindung koennen genutzt werden
asyncmap 0
# benutze eine maximum transmission packet Groesse von 552 Bytes
mtu 552
# benutze eine maximum receive packet Groesse von 552 Bytes
mru 552
```

12 Konfiguration der `/etc/resolv.conf` Datei

Während wir Menschen es lieben, den Dingen Namen zu geben, mögen Computer viel lieber Nummern. In einem TCP/IP Netzwerk sprechen wir die Rechner mit einem bestimmten Namen an, wobei jede Maschine in einem bestimmten Domain zu finden ist. Zum Beispiel hat meine Linux Workstation den Namen `archenland` und ist in dem `hedland.edu.au` Domain angesiedelt. Die für einen Menschen gut zu lesende und zu merkende Adresse ist also `archenland.hedland.edu.au`. Damit die anderen Rechner diesen Computer im Internet finden und ansprechen können, ist er außerdem unter seiner IP Nummer (203.19.8.7) bekannt.

Das Übersetzen (resolving) des Maschinen und Domain Names in die im Internet verwendeten IP Nummern ist die Aufgabe einer Maschine, die den Domain Name Service anbietet. Die Rechner wird meistens auch als Nameserver bezeichnet.

Wenn man also eine PPP Verbindung aufbaut, muß man dem eigenen Linux Rechner irgendwie mitteilen, welcher Rechner die Hostnamen in IP Nummern umwandelt, so daß man selbst die Maschinen Namen verwenden kann, der Rechner diese aber in IP Nummern umwandeln kann.

Eine Möglichkeit wäre, alle Hosts, mit denen man kommunizieren möchte, in die `/etc/hosts` Datei einzutragen. Diese Möglichkeit findet aber nur in lokalen LANs mit wenigen Rechner Anwendungen. Im Internet scheidet diese Möglichkeit wegen der riesigen Anzahl vom Rechner aus. Als zweite Möglichkeit könnte man selbst die Hosts nicht mehr durch ihren Namen in Programmen eingeben, sondern die IP Nummern direkt verwenden. Allerdings kann man sich sehr schlecht mehr als ein paar dieser Nummern merken, so daß auch diese Möglichkeit nur für sehr kleine LANs angewendet wird.

Der beste Weg ist es, Linux so zu konfigurieren, daß es weiß, woher es diese Informationen für die Namen nach Nummer Umsetzung automatisch herbekommen kann. Dieser Service wird vom Domain Name Server System angeboten. Hierfür muß man nur die IP Nummer eines solchen DNS Systems in der `/etc/resolv.conf` Datei angeben.

Die meisten Provider bieten zwei DNS Server für ihre Kunden an. Es ist zwar nur einer notwendig, aber man setzt zur Sicherheit meistens zwei DNS Server ein, falls einer mal ausfallen sollte.

Ihre `/etc/resolv.conf` sollte wie folgende aussehen:

```
domain ihr.isp.domain.name
nameserver 10.25.0.1
```

Diese Datei muß an den eigenen Provider angepaßt werden. Hierfür sollte der Provider die benötigten Informationen bereitstellen. Die Datei sollte folgende Besitzer und Zugriffsrechte haben.

```
-rw-r--r--   1 root     root           73 Feb 19 01:46 /etc/resolv.conf
```

Falls Sie bereits die `/etc/resolv.conf` eingerichtet haben, weil Ihr Rechner mit einem LAN verbunden ist, fügen Sie einfach die IP Nummer des PPP DNS Server zu den existierenden hinzu.

13 PPP Verbindung manuell aufbauen

Nachdem man nun eigene `/etc/ppp/options` und `/etc/resolv.conf` Dateien erzeugt hat, kann man die Einstellungen testen, in dem man per Hand eine PPP Verbindung aufbaut. Wenn der manuelle Verbindungsaufbau erst einmal funktioniert, werden wir ihn automatisieren.

Um den manuellen Verbindungsaufbau durchzuführen, benötigt man ein Terminalprogramm, das beendet werden kann, ohne das Modem zu resetten. Minicom kann dieses mittels `ALT-Q` (oder in alten Versionen von `minicom` mittels `CTRL-A-Q`).

Stellen Sie dabei sicher, daß sie als `root` eingeloggt sind.

Starten Sie nun das Terminalprogramm (wie z.B. `minicom`), wählen Sie sich in den PPP Server ein und loggen Sie sich ganz normal ein. Falls Sie einen Befehl auf dem Server eingeben müssen, um PPP zu starten, tun Sie dieses. Sie sehen nun den Müll auf den Bildschirm, den Sie vorher schon gesehen haben.

Man verläßt nun das Terminalprogramm ohne das Modem zu resetten (`ALT-Q` in `minicom`) und gibt folgendes als `root` am Linux Prompt ein:

```
pppd -d -detach /dev/cuaX &
```

(für dynamische IP Nummern) oder

```
pppd -d -detach <Ihre IP Nummer>: /dev/cuaX
```

(für statische IP Nummern - beachten Sie den Doppelpunkt nach der IP Nummer in diesem Fall)

Die `-d` Option schaltet den Debugging Modus ein. Hierdurch wird die "Konversation"während des Verbindungsaufbaus in der Logdatei des Systems festgehalten. Dieses ist nützlich, falls es Probleme geben sollte.

Die Status LEDs des Modems sollten nun blinken, wenn die PPP Verbindung aufgebaut wird. Es dauert einen kurzen Augenblick, bis die Verbindung aufgebaut ist.

Zu diesem Zeitpunkt kann man sich das PPP Interface anschauen, indem man folgenden Befehl eingibt

```
ifconfig ppp0
```

Man sollte nun ungefähr folgendes sehen können:

```
ppp0      Link encap:Point-Point Protocol
          inet addr:10.144.153.104  P-t-P:10.144.153.51 Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:552  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

Wobei

- `inet addr:10.144.153.104` die eigene IP Nummer und
- `P-t-P:10.144.153.51` die IP Nummer des Servers ist.

Natürlich wird `ifconfig` nicht genau diese IP Nummer anzeigen, sondern diejenigen, die Ihr PPP Server benutzt.

Hinweis: `ifconfig` zeigt außerdem an, daß die Verbindung UP und RUNNING ist!

Falls man etwa sowas erhält

```
ppp0      Link encap:Point-Point Protocol
          inet addr:0.0.0.0  P-t-P:0.0.0.0  Mask:0.0.0.0
          POINTOPOINT  MTU:1500  Metric:1
          RX packets:63287 errors:0 dropped:0 overruns:0
          TX packets:62187 errors:0 dropped:0 overruns:0
```

hat der Aufbau der PPP Verbindung nicht funktioniert. Lesen Sie dann den späteren Abschnitt über Debugging.

Nun kann man die Verbindung testen, in dem man die IP Nummer des Servers, die `ifconfig` angezeigt hat, anpingt, z.B.

```
ping 10.144.153.51
```

Man sollte eine solche Ausgabe erhalten

```
PING 10.144.153.51 (10.144.153.51): 56 data bytes
64 bytes from 10.144.153.51: icmp_seq=0 ttl=255 time=328.3 ms
64 bytes from 10.144.153.51: icmp_seq=1 ttl=255 time=190.5 ms
64 bytes from 10.144.153.51: icmp_seq=2 ttl=255 time=187.5 ms
64 bytes from 10.144.153.51: icmp_seq=3 ttl=255 time=170.7 ms
```

Diese Anzeige läuft weiter, bis man sie mittels `CTRL-C` abbricht. Man erhält dann noch einige weitere Informationen:

```
--- 10.144.153.51 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 170.7/219.2/328.3 ms
```

So weit so gut.

Nun sollte man einen Rechner über seinen Namen anpingen, wobei dieses nicht der PPP Server selbst sein sollte. Stattdessen verwendet man einen Rechner, von dem man weiß, daß er läuft, z.B.:

```
ping hedunx.hedland.edu.au
```

Dieses mal gibt es eine kurze Pause, bis die erste Ausgabe von `ping` erscheint. Dieses liegt daran, daß Linux ja zuerst die IP Nummer von dem DNS Server holen muß, den man in der `/etc/resolv.conf` angegeben hat. Man erhält dann eine solche Ausgabe:

```
 PING hedunx.hedland.edu.au (10.144.153.3): 56 data bytes
64 bytes from 10.144.153.3: icmp_seq=0 ttl=254 time=190.1 ms
64 bytes from 10.144.153.3: icmp_seq=1 ttl=254 time=180.6 ms
64 bytes from 10.144.153.3: icmp_seq=2 ttl=254 time=169.8 ms
64 bytes from 10.144.153.3: icmp_seq=3 ttl=254 time=170.6 ms
64 bytes from 10.144.153.3: icmp_seq=4 ttl=254 time=170.6 ms
```

Und wieder bricht man die Ausgabe mit `CTRL-C` ab und erhält die Statistik:

```
--- hedunx.hedland.edu.au ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 169.8/176.3/190.1 ms
```

Falls man überhaupt keine Antwort erhalten sollte, findet man ein Erklärung in dem Debugging Abschnitt dieses Dokuments.

Wenn alles läuft, kann man die Verbindung jetzt wieder mit folgendem Befehl abbauen:

```
ppp-off
```

Falls das nicht funktionieren sollte, schalten man entweder das Modem ab oder man startet das Terminalprogramm, unterbricht das Modem mit +++ und legt dann mit `ATH0` auf, wenn man das `OK` von Modem erhalten hat.

14 Automatisierung der Verbindung mit chat

Ein `chat` Skript automatisiert den Einlogvorgang, so daß man als `root` oder als Mitglied der `ppp` Gruppe nur einen einzelnen Befehl eingeben muß, um eine Verbindung aufzubauen.

Ein `chat` Skript besteht jeweils aus Paaren von einer Zeichenkette, die `chat` erwartet und die Antwort die er darauf senden soll Nachdem das Modem zum Beispiel gewählt hat und eine Verbindung zum anderen Modem aufgebaut hat, warten wir darauf, das Login Prompt zu empfangen, wobei dieses eventuell erst nach einer Menge Text aus der `/etc/issue.net` Datei des Servers folgt. Nachdem wir dieses empfangen haben, senden wir unseren Benutzernamen. Als nächstes warten wir auf das Paßwort Prompt und senden unser Paßwort.

Im Prinzip also ganz einfach.

Falls das PPP Paket korrekt installiert worden ist, sollten sich in `/usr/sbin` zwei Dateien mit folgenden Namen befinden:

```
ppp-on
ppp-off
```

Diese enthalten Templates für den Verbindungsaufbau einer PPP Verbindung unter Verwendung von `chat`. Die `ppp-off` Datei war diejenigen, die wir im letzten Abschnitt verwendet haben, um unsere manuell aufgebaute PPP Verbindung zu beenden.

Nur für den Fall, daß bei Ihnen diese Templates fehlen sollten, sind hier modifizierte und kommentierte Versionen der Dateien, die wir verwenden werden:

/usr/sbin/ppp-on:

```
#!/bin/sh
#
#       ppp-on
#
#       Baut eine PPP Verbindung auf
#

LOCKDIR=/var/lock   # Verzeichnis des lock files
DEVICE=cua1         # Verwendete serielle Schnittstelle
PHONE=XXXXXXXX      # Telefonnummer des Modems des ISPs
USER=XXXX           # Benutzername zum Einloggen
PASSWORD=password   # Passwort beim Einloggen
OUR_IP_ADDR=x.x.x.x # Die eigene IP Nummer bei der Verwendung von
                    # statischen IP Nummern, ansonsten 0.0.0.0

if [ -f $LOCKDIR/LCK..$DEVICE ]
then
   echo "PPP device ist gelockt"
   exit 1
fi

fix-cua $DEVICE
(
   stty 19200 -tostop

   # Dieses kann man auch 115200 aendern, wenn die serielle Schnittstelle
   # des Modems dieses verkraftet.

   if chat -v -l LCK..$DEVICE ABORT "NO CARRIER" ABORT BUSY "" ATZ OK\
      ATDT$PHONE CONNECT "" ogin: $USER ssword: \\q$PASSWORD
   then
       pppd -d -detach asyncmap 0 modem crtscts $OUR_IP_ADDR: /dev/$DEVICE
       rm -f LCK..$DEVICE
       exit 0
   else
```

```
            echo "PPP Anruf fehlgeschlagen" 1>&2
            exit 1
        fi
    ) < /dev/$DEVICE > /dev/$DEVICE
```

/usr/sbin/ppp-off:

```
#!/bin/sh

DEVICE=ppp0

# Falls die PID Datei von ppp0 existiert, laeuft der Daemon.
# Beende ihn.

if [ -r /var/run/$DEVICE.pid ]; then
       kill -INT `cat /var/run/$DEVICE.pid`

# Falls nicht erfolgreich stelle sicher, dass die PID Datei
# entfernt wird.

       if [ ! "$?" = "0" ]; then
               echo "Entferne stale $DEVICE PID Datei."
               rm -f /var/run/$DEVICE.pid
               exit 1
       fi
#
# Erfolg. Beendet mit richtigem Status.
#
       echo "$DEVICE link beendet"
       exit 0
fi

#
# Der link ist nicht aktiv
#

echo "$DEVICE link ist nicht aktiv"
exit 1
```

Die Rechte der Dateien und ihre Besitzer sollten sein:

```
-rwxr-xr-x   1 root     PPP          502 Sep  3 20:04 /usr/sbin/ppp-off
-rwxr-xr-x   1 root     PPP          612 Sep  3 20:04 /usr/sbin/ppp-on
```

14.1 Was ein chat Skript bedeutet...

Ein `chat` Skript ist eine Folge von erwarte und sende Strings. Hierbei sollte man beachten, daß wir *immer* zuerst etwas erwarten, bevor wir etwas senden. Wenn wir etwas senden möchten, ohne vorher etwas zu empfangen, müssen wir einen leeren erwarte String verwenden (gekennzeichnet durch `""`) und das gleiche gilt für das Erwarten einer Zeichenkette, ohne das etwas gesendet werden soll. Auch wenn eine Zeichenkette aus mehreren Worten wie z.B. `NO CARRIER` besteht, muß man die Zeichenkette in Anführungszeichen einschließen.

Die chat Zeile in unserem Beispiel ist:

```
if chat -v -l LCK..$DEVICE ABORT "NO CARRIER" ABORT BUSY "" ATZ OK\
   ATDT$PHONE CONNECT "" ogin: $USER ssword: \\q$PASSWORD
```

Dieses bedeutet ...

- `chat -v -l LCK..$DEVICE`
  rufe `chat` auf und erzeuge eine lock Datei für diese serielle Schnittstelle, damit nur diese Applikation auf die Schnittstelle zugreift (`$DEVICE` ist eine Shellvariable, die die Schnittstelle festlegt). Das `-v` fordert `chat` auf, all sein I/O in der Logdatei des Systems mitzuschreiben(normalerweise `/var/log/messages`)
- `ABORT NO CARRIERÄBORT BUSY`
  Falls Du eine dieser Antworten erhälst, breche den Versuch ab.
- `ATZ`
  erwarte nichts (), sende dann `ATZ` (setzt das Modem zurück).
- `OK ATDT$PHONE`
  erwarte `OK` (die Modem Antwort auf `ATZ`), wähle `$PHONE` (eine andere Shellvariable, die die Nummer angibt, die gewählt werden soll).
- `CONNECT`
  erwarte CONNECT (das Modem auf der anderen Seite hat geantwortet und die Modems haben erfolgreich die Baud Rate usw. abgestimmt), sende nichts ().
- `ogin: $USER`
  erwarte `ogin:`, das Login Prompt (ohne das führende `l` - dieses muß an das Prompt angepaßt werden, das der eigene ISP benutzt), und sende `$USER` (eine Shellvariable, die den eigenen Namen enthält).
- ssword: \\q$PASSWORD
  erwarte `assword:` das Passwort Prompt (ohne das führende `p`) und sende `$PASSWORD` (eine weitere Shellvariable, die das eigene Paßwort enthält) Das `\\q` fordert `chat` auf, daß Paßwort *nicht* mittels syslog zu speichern, wenn es mit `-v` aufgerufen wird.

Dieses ist ein sehr einfaches `chat` Skript. `chat` verfügt zusätzlich über einer Fehlerbehandlung und eine Anzahl weiterer Optionen. Für weitere Informationen lesen Sie bitte die chat manual page (man 8 chat).

15 Erzeugen einer eigenen `ppp-on` Datei

Kopieren Sie die `ppp-on` Datei unter einem neuen Namen. Der Autor selbst benutzt `ppp-providername`. Auf diese Weise kann man immer wieder mit dem Original anfangen, falls es zu Problemen mit der selbst erstellten Version kommen sollte.

Nun werden wir die Datei editieren, wobei wir die Informationen verwenden, die wir vom System Administrator des PPP Servers erhalten haben.

`ppp-on` ist ein Bourne Shell Skript. Als erstes setzt es eine Anzahl von Shellvariablen für die spätere Verwendung.

15.1 Lock file Verzeichnis (LOCKDIR)

Suchen Sie in dem Verzeichnis `/var` und dessen Unterverzeichnissen nach Ihrem lock file Verzeichnis. Für ein Linux System nach dem *FSSTND* sollte dieses `/var/lock` sein. Ältere Systeme benutzen eventuell `/var/spool/lock`.

Ändern Sie die `LOCKDIR=` Zeile so, daß sie auf das lock Verzeichnis Ihres Systems zeigt.

15.2 Serielle Schnittstelle (DEVICE)

Dieses ist die serielle Schnittstelle, die mit dem zu verwendenden Modem verbunden ist, wobei das `/dev/` Prefix weggelassen wird.

Editieren sie die `DEVICE=` Zeile, so daß sie auf die serielle Schnittstelle zeigt, an dem das Modem hängt.

15.3 PPP Server Telefonnummer (PHONE)

Dieses ist die Telefonnummer, die gewählt werden muß, um den PPP Server zu erreichen.

Editieren Sie die `PHONE=`, so daß sie die Telefonnummer des PPP Servers enthält.

15.4 Benutzername (USER)

Dieses ist der Benutzername, der zum Einloggen in den PPP Server verwendet wird.

Editieren Sie die `USER=` Zeile, so daß sie Ihren Benutzernamen auf dem PPP Server enthält. Dabei sollte man bedenken, daß Unix und die meisten PPP Server die Klein- und Großschreibung unterscheiden.

15.5 Paßwort (PASSWORD)

Dieses ist das Paßwort, das man zum Einloggen in den PPP Server verwendet.

Editieren sie die `PASSWORD` Zeile, so daß sie Ihr Paßwort enthält. Auch hier wird meistens die Klein- und Großschreibung unterschieden.

Nun gibt es ein kleines Problem. Mit den Zugriffsrechten, wie sie gerade gesetzt sind, kann jeder die `ppp-on` Datei lesen und kann so an das Paßwort gelangen. Aus diesem Grund muß man die Dateirechte nach dem Editieren unbedingt auf 750 ändern (`chmod 750 /usr/sbin/ppp-on`), so daß nur `root` und die Mitglieder der `ppp` Gruppe diese Datei lesen können.

Denken Sie bitte daran, daß, auch wenn Sie der einizige Benutzer des Rechners sind, Sie eine Verbindung zum Internet oder einem anderen Netzwerk aufbauen. Da Linux ein Multiuser Betriebssystem ist, könnte jemand versuchen, sich in Ihr System einzuloggen, während die Verbindung besteht und an Ihre Daten (Name und Paßwort) für Ihren PPP Account gelangen. Mit diesen Daten hätte der Angreifer dann die Möglichkeit, sich auf Ihre Kosten ins Internet einzuwählen und unter Ihrem Namen Straftaten zu begehen.

15.6 Ihre IP Adresse (OUR_IP_ADDR)

Falls der Server, zu dem man eine Verbindung aufbauen möchte, *dynamische* IP Nummern verwendet, muß dieses auf 0.0.0.0 gesetzt werden. Noch besser ist es allerdings, diese Zeile dann komplett zu löschen und auch das `$OUR_IPADDR:` vom `pppd` Aufruf zu entfernen.

Falls Sie vom Administrator des PPP Servers eine eigene IP Nummer erhalten haben, benutzen Sie *statische* IP Nummern. In diesem Fall muß diese IP Nummer in dieser Zeile eingetragen werden.

15.7 Editieren der `chat` Zeile

Der gesamte `chat` Befehl muß aus einer Zeile bestehen. Zeilenumbrüche sind nicht erlaubt. Ausnahmen werden in der man page von `chat` beschrieben.

Die `chat` Zeile beginnt mit Optionen für `chat` selbst.

```
-l LCK..$DEVICE
```

Dieses richtet den lock file ein.

Eine andere *sehr* nützliche Option für das Debuggen ist `-v`, welche `chat` auffordert, den gesamten Einlogvorgang mittels syslog mitzuspeichern. Wenn man das erste mal eine Verbindung aufbaut, sollte man diese Option verwenden, um eventuelle Fehler aufspüren zu können. Der Anfang der `chat` Zeile wäre also

```
-v -l LCK..$DEVICE
```

Die erwarte/sende Paare beginnen mit

```
"" ATZ
```

Dieses bedeutet, erwarte nichts und sende `ATZ`, welches das Modem zurücksetzt und dessen Standardeinstellung lädt.

Nun haben wir

```
OK ATDT$PHONE
```

welches auf die `OK` Antwort des Modems als Antwort auf den `ATZ` Befehl wartet und dann das Kommando für die Tonwahl und die entsprechende Telefonnummer sendet. Falls es nicht gelingt, daß Modem so zu konfigurieren, daß es nach dem Reset die richtigen Einstellungen für die PPP Verbindung verwendet, kann man hier einen String für die Initialisierung des Modems einbinden. Zum Beispiel

```
OK ATDT&K3S0=0DT$PHONE
```

Das nächste Paar ist

```
CONNECT ""
```

welches bedeutet, erwarte `CONNECT`, welches das Modem nach einem Verbindungsaufbau mit dem anderen Modem sendet, und sende nichts.

Nun kommen wir zum eigentlich Einlogprozeß

```
ogin: $USER
```

welches auf das login prompt wartet und dann den eigenen Benutzernamen sendet. Der String, der von `chat` erwartet wird, muß an den Loginprozeß des eigenen Providers, den man vorher genau mitgeschrieben hat, angepaßt werden.

Nun warten wir auf die Aufforderung, daß Paßwort einzugeben

```
assword: \\q$PASSWORD
```

und senden unser Paßwort. Das `\\q` fordert `chat` auf, daß Paßwort nicht in den Logdateien des Systems zu speichern, was ansonsten bei Verwendung der `-v` Option passieren würde.

Falls Sie auf dem PPP Server zuerst ein Kommando eingeben müssen, um PPP zu starten, fügen Sie ein zusätzliches erwarte/sende Paar hinzu. Dieses sollte auf das Login Prompt warten, wobei man auf bestimmte Buchstaben wie $ und [oder] achten sollte, die beim Bourne Shell eine besondere Bedeutung haben, und den Befehl zum Start von PPP senden.

In meinem Fall benutzt der PPP Server das normale Linux Bash Prompt

```
[hartr@kepler hartr]$
```

und erwartet, daß ich folgendes eingebe

```
ppp
```

um PPP auf dem Server zu starten, so daß mein erwarte/sende Paar so aussieht

```
hartr ppp
```

Das `chat` Programm hat noch viel mehr Funktionen, wie wir hier beschrieben haben, wie z.B. die Fehlerbehandlung und fehlende erwarte Strings. Weitere Informationen zu `chat` befinden sich in dessen man page.

Manche PPP Server haben nicht bei jedem Einloggen genau die gleichen Eingabeaufforderungen, so daß man sich eventuell mehrere male per Hand einloggen muß, um zu verstehen, was genau passiert und welche erwarte Strings man verwenden kann.

15.8 Editieren der `pppd` Zeile

Da man ja eine `/etc/ppp/options` Datei eingerichtet hat, muß man dem `pppd` keine weiteren Optionen außer `$DEVICE` übergeben. Diese Option sagt dem `pppd`, welche serielle Schnittstelle er verwenden soll.

Allerdings kann man auch alle Optionen auf der Kommandozeile übergeben, wobei der `pppd` Befehl und alle seine Optionen in einer Zeile stehen müssen und das device gefolgt von der Geschwindigkeit der Schnittstelle die letzten übergebenen Optionen sein müssen.

Wie wir bereits gesehen haben, kann das Loggen der Debuginformationen mit der Option `-d` vom `pppd` eingeschaltet werden. Dieses sollte man bei den ersten Versuchen mit einem neuen Skript immer verwenden. Allerdings sollte man, wenn man nur über wenig freien Platz auf der Festplatte verfügt, bedenken, daß die Logdateien einigen Platz auf der Festplatte einnehmen können.

Falls man die Datei mit den PPP Optionen nicht `/etc/ppp/options` oder `/etc/ppp/options.ttySx` genannt hat, übergibt man den Dateinamen mit der `-f` Option an den `pppd`, also z.B.

```
-f options.myserver
```

15.9 Speichern des Skriptes

Das war es. Verlassen Sie den Editor und ändern sie die Zugriffsrechte der Skript Datei auf 750!

16 Testen des chat Skriptes

Öffnen Sie ein neues `root` Xterm, falls Sie sich in X befinden, oder öffnen Sie eine neue virtuelle Konsole und loggen Sie sich als `root` ein.

Hier geben Sie nun folgendes Kommando ein

```
tail -f /var/log/messages
```

Wobei Sie `/var/log/messages` eventuell durch den Namen der Datei ersetzen müssen, in der der `syslogd` Meldungen vom Typ *Daemon* speichert.

In dem ersten Fenster oder der ersten virtuellen Konsole geben Sie folgendes Kommando ein

```
ppp-on &
```

oder wie Sie Ihre editierte Version von `/usr/sbin/ppp-on` genannt haben. Falls Sie das Skript nicht mittels `&` am Ende des Kommandos in den Hintergrund starten, bekommen Sie kein Terminal Prompt zurück, bevor die PPP Verbindung beendet wird.

Nun schalten Sie auf das andere Fenster zurück, das Ihre Logdatei anzeigt.

Nun werden Sie etwas wie das folgende sehen, falls Sie `chat` mit `-v` und `pppd` mit `-d` aufgerufen haben. Dieses zeigt die Arbeit des `chat` Skriptes beim Einloggen und die Startinformationen des `pppd`.

```
Oct 21 16:09:58 hwin chat[19868]: abort on (NO CARRIER)
Oct 21 16:09:59 hwin chat[19868]: abort on (BUSY)
Oct 21 16:09:59 hwin chat[19868]: send (ATZ^M)
Oct 21 16:09:59 hwin chat[19868]: expect (OK)
Oct 21 16:10:00 hwin chat[19868]: ATZ^M^M
Oct 21 16:10:00 hwin chat[19868]: OK -- got it
Oct 21 16:10:00 hwin chat[19868]: send (ATDT722298^M)
Oct 21 16:10:00 hwin chat[19868]: expect (CONNECT)
Oct 21 16:10:00 hwin chat[19868]: ^M
Oct 21 16:10:22 hwin chat[19868]: ATDT722298^M^M
Oct 21 16:10:22 hwin chat[19868]: CONNECT -- got it
Oct 21 16:10:22 hwin chat[19868]: send (^M)
Oct 21 16:10:22 hwin chat[19868]: expect (ogin:)
Oct 21 16:10:22 hwin chat[19868]:  57600^M
Oct 21 16:10:23 hwin chat[19868]: ^[[;H^[[2J^M^M
Oct 21 16:10:23 hwin chat[19868]: ^M
Oct 21 16:10:23 hwin chat[19868]: ^M
Oct 21 16:10:23 hwin chat[19868]: ^I^I   This is node  kepler.hedland.edu.au^M
Oct 21 16:10:23 hwin chat[19868]: ^I^I^I      at Hedland Campus^M
Oct 21 16:10:23 hwin chat[19868]: ^I^I^I       Hedland College^M
Oct 21 16:10:23 hwin chat[19868]: ^M
Oct 21 16:10:23 hwin chat[19868]: ^I^I  Authorised user ON-
LY are to use this system^M
Oct 21 16:10:23 hwin chat[19868]: ^M
Oct 21 16:10:23 hwin chat[19868]: ^M
Oct 21 16:10:23 hwin chat[19868]: ^I^I For more information, con-
tact ComputerSystems^M
Oct 21 16:10:23 hwin chat[19868]: ^I^I^I     on +61 (0)91 72 0400^M
Oct 21 16:10:23 hwin chat[19868]: ^I^I^I^I     or^M
Oct 21 16:10:23 hwin chat[19868]: ^I^I       email:  help@hedunx.hedland.edu.au^M
Oct 21 16:10:23 hwin chat[19868]: ^M
Oct 21 16:10:23 hwin last message repeated 3 times
Oct 21 16:10:23 hwin chat[19868]: kepler login: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (hartr^M)
Oct 21 16:10:23 hwin chat[19868]: expect (ssword:)
Oct 21 16:10:23 hwin chat[19868]:  hartr^M
Oct 21 16:10:23 hwin chat[19868]: Password: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (??????^M)
Oct 21 16:10:23 hwin chat[19868]: expect (hartr)
Oct 21 16:10:23 hwin chat[19868]:  ^M^M
Oct 21 16:10:24 hwin chat[19868]: Last login: Sat Oct 21 14:55:53 on  ttyC0^M
Oct 21 16:10:24 hwin chat[19868]: ^M
Oct 21 16:10:24 hwin last message repeated 9 times
Oct 21 16:10:24 hwin chat[19868]: ^I^IYou have logged into no-
de  kepler.hedland.edu.au^M
```

```
Oct 21 16:10:24 hwin chat[19868]: ^M
Oct 21 16:10:24 hwin chat[19868]: ^M
Oct 21 16:10:24 hwin chat[19868]: [hartr -- got it
Oct 21 16:10:24 hwin chat[19868]: send (ppp^M)
Oct 21 16:10:27 hwin pppd[19872]: pppd 2.1.2 started by root, uid 0
Oct 21 16:10:27 hwin pppd[19873]: Using interface ppp0
Oct 21 16:10:27 hwin pppd[19873]: Connect: ppp0 <--> /dev/cua1
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(LCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: LCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MRU
Oct 21 16:10:27 hwin pppd[19873]: (1500)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ASYNCMAP
Oct 21 16:10:27 hwin pppd[19873]: (0)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MAGICNUMBER
Oct 21 16:10:27 hwin pppd[19873]: (a098b898)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd PCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ACCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: returning CONFACK.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(LCP): Sent code 2, id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfack(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:27 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:27 hwin pppd[19873]: (45)
Oct 21 16:10:27 hwin pppd[19873]:  (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:28 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 1.
Oct 21 16:10:30 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:30 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:30 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 255.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:31 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:31 hwin pppd[19873]:  (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:31 hwin pppd[19873]: (45)
Oct 21 16:10:31 hwin pppd[19873]:  (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:31 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 255.
Oct 21 16:10:31 hwin pppd[19873]: fsm_rconfack(IPCP): Rcvd id 1.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: up
Oct 21 16:10:31 hwin pppd[19873]: local  IP address 10.144.153.104
Oct 21 16:10:31 hwin pppd[19873]: remote IP address 10.144.153.51
```

Der Autor verwendet *statische* IP Nummern, daher sendet der Rechner dieses an den PPP Server. Dieses werden Sie, wenn Sie *dynamische* verwenden, nicht sehen.

Dieses sieht gut aus, testen Sie nun, wie vorher beschrieben, ob Sie IP Nummern und Hostnamen anpingen können.

Sie können nun Ihren WWW Browser oder was auch immer starten. Sie sind nun mit dem Internet verbunden.

17 Abbauen der PPP Verbindung

Wenn die PPP Verbindung nicht mehr länger benötigt wird, kann sie einfach mit dem `ppp-off` Befehl abgebaut werden. Auch hierfür muß man `root` oder Mitglied der `ppp` Gruppe sein.

In Ihrer Logdatei wird dann ungefähr folgendes zu finden sein

```
Oct 21 16:10:45 hwin pppd[19873]: Interrupt received: terminating link
Oct 21 16:10:45 hwin pppd[19873]: ipcp: down
Oct 21 16:10:45 hwin pppd[19873]: default route ioctl(SIOCDELRT): Bad address
Oct 21 16:10:45 hwin pppd[19873]: fsm_sdata(LCP): Sent code 5, id 2.
Oct 21 16:10:46 hwin pppd[19873]: fsm_rtermack(LCP).
Oct 21 16:10:46 hwin pppd[19873]: Connection terminated.
Oct 21 16:10:46 hwin pppd[19873]: Exit.
```

18 Ich bekomme keine Verbindung ...

Es gibt eine große Anzahl von Gründen, warum dieses passieren kann. `chat` ist es nicht gelungen, sich einzuloggen oder die Telefonverbindung ist zu schlecht. Genauere Informationen erhält man aus dem syslog.

In einem solchen Fall sollte man einen Blick in die *PPP FAQ* werfen, die eine wirklich große Anzahl von Fragen und Antworten enthält. Dieses ist ein sehr kompaktes Dokument und die Antwort ist dort zu finden. Aus meiner eigenen Erfahrung kann ich sagen, daß, wenn die Antwort für das Problem dort nicht zu finden ist, das Problem nicht der Fehler von ppp ist. In meinem Fall habe ich ein ELF Kernel benutzt und keine neue Version eingespielt, als ich die neue Kernel Modules installiert habe.

19 Nachdem die Verbindung aufgebaut ist ...

19.1 Spezielles Routing, Leeren der sendmail Warteschlange usw.

Ein weiterer Schritt der Automatisierung ist möglich. Sobald die PPP Verbindung aufgebaut ist, sucht der `pppd` nach `/etc/ppp/ip-up`. Falls dieses Skript existiert und es ausführbar ist, führt der PPP Daemon das Skript aus. Dieses ermöglicht es, nach jedem Verbindungsaufbau spezielle Kommandos ausführen zu lassen.

Hiermit kann man also z.B. `sendmail` dazu bringen, die wartenden emails zu verschicken. Außerdem kann man hier Kommandos einfügen, die emails und News vom ISP abholen.

20 Beenden der Verbindung

Das existierende `/usr/sbin/ppp-off` Skript sollte einwandfrei funktionen. Die einzig sinnvolle Änderung könnte darin bestehen, daß Skript so zu ändern, daß es wartet, bis `sendmail` alle emails abgeliefert hat.

Zusätzlich kann man ein Skript erzeugen, daß immer dann ausgeführt wird, wenn die Verbindung beendet worden ist. Dieses wird in der Datei `/etc/ppp/ip-down` gespeichert. Es kann dazu verwendet werden, spezielle Einstellung zurückzusetzen, die vorher mit dem korrespondierenden `/etc/ppp/ip-up` Skript gesetzt worden sind.

21 Routing in einem LAN

Falls der eigene Rechner nicht nur per PPP mit dem Internet verbunden werden soll, sondern gleichzeitig auch noch eine Verbindung zum lokalen LAN über die Ethernetkarte bestehen soll, muß man dem Rechner mitteilen, welche Rechner er über welches Interface erreicht.

Dieser Abschnitt versucht *nicht*, das Routen zu vermitteln. Vielmehr beschäftigt er sich mit ganz spezielle statischen Routen.

Ich würde Ihnen dringend empfehlen, daß Linux *Network Administrator Guide* (NAG) zu lesen, wenn Sie sich mit dem Routen bisher nicht auskennen. Auch das O'Reilly Buch *TCP/IP Netzwerk Administration* beschreibt dieses Thema in einer leicht verständlichen Form.

Die Grundregel vom statischen Routen lautet, daß die *default* route immer diejenige sein sollte, die auf die größte Anzahl von Netzwerkadressen zeigt. Für andere Netzwerk legt man spezielle Routen in der Routingtabelle fest.

Die einzige Situation, die ich hier beschreiben werden, ist der Fall, wo sich der eigene Linux Rechner in einem LAN befindet, das nicht mit dem Internet verbunden ist und man sich in das Internet einwählen möchte, während man weiterhin mit dem LAN verbunden ist.

Als erstes sollte man sicherstellen, daß die Ethernet Route auf die spezielle Netzwerkadressen zeigt, die in dem eigenen LAN vorhanden sind und nicht auf die default route.

Dieses kann man mit dem `route` Befehl überprüfen, wobei man etwa folgendes sehen sollte:

```
[root@hwin /root]# route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
loopback        *               255.255.255.0   U     1936   0        50 lo
10.0.0.0        *               255.255.255.0   U     1436   0       565 eth0
```

Falls das Ethernet Interface (`eth0`) auf die default route zeigt, was man an einem `default` in der erste Spalte der `eth0` Zeile erkennen kann, muß man die Ethernet Initialisierungsskripte ändern, so daß es auf spezielle Netzwerknummer zeigt. Weitere Informationen hierzu finden sich in der *NET3 HOWTO* und im *NAG*.

Dieses wird es dem `pppd` ermöglichen, die default route, wie unten gezeigt, zu setzen:

```
[root@hwin /root]# route -n
Kernel routing table

Destination     Gateway         Genmask         Flags MSS    Window Use Iface
10.144.153.51   *               255.255.255.255 UH    488    0         0 ppp0
127.0.0.0       *               255.255.255.0   U     1936   0        50 lo
10.1.0.0        *               255.255.255.0   U     1436   0       569 eth0
default         10.144.153.51   *               UG    488    0         3 ppp0
```

Wie man sehen kann, haben wir eine host route zum PPP Server (10.144.153.51) über `ppp0` und außerdem eine default network route, die den PPP Server als Gateway verwendet.

Falls Ihr Netzwerk kompilizierter als das hier gezeigte aussehen sollte, lesen Sie die schon oben erwähnten Dokumente über das Routing oder fragen Sie einen Experten in Ihrem Netzwerk.

Falls sich in Ihrem LAN bereit Router befinden, haben Sie bereits eingerichte Gateways zu den anderen Netzwerken Ihrer Site. Die default route sollte aber trotzdem auf der PPP Interface zeigen. Für die anderen Netzwerke kann man spezielle Routen anlegen.

21.1 Hinweise zur Sicherheit

Wenn man sich mit dem eigenen Linux Rechner, der bereits mit einem LAN verbunden ist, in das Internet einwählt, öffnet man das gesamte LAN für das Internet und die eventuell dort zu finden Hacker. Aus diesem Grund sollte der Verbindungsaufbau mit dem Internet auf jeden Fall mit dem Netzwerk Administrator des LANs und dem Sicherheitsbeauftragen abgestimmt werden. Ansonsten kann man sich bei einem erfolgreichen Einbruch von Hackern in das lokale Netzwerk und dessen Rechner ziemlichen Ärger einhandeln.

Bevor man ein LAN mit dem Internet verbindet, sollte man sich zuerst Gedanken über die dadurch entstehenden Sicherheitsrisiken machen. Dieses gilt auch bei einer dynamischen Verbindung.

22 Debuggen

Falls es einfach nicht gelingt, die PPP Verbindung zum korrekten Arbeiten zu bringen, sollte Sie dieses Dokument noch einmal genau lesen und alles überprüfen und zwar in Verbindung mit der von `chat -v` und `pppd -d` erzeugten Ausgabe in der Logdatei des Systems.

Auch sollten Sie die PPP Dokumentation und die FAQ zusammen mit den hier erwähnten Dokumenten zu Rate ziehen.

Falls auch das nicht helfen sollte, beschreiben Sie Ihr Problem in den Newsgruppen `de.comp.os.linux.networking` und `comp.protocols.ppp`. Hier lesen eigentlich immer Leute mit, die Ihnen helfen können. Man sollte jedoch immer daran denken, nur die wichtigsten Zeilen aus der Logdatei zu posten. Mails mit etlichen Kilobytes Logdateien werden nicht gerne gesehen und meistens auch nicht beantwortet.

23 Probleme, wenn die Verbindung erst einmal arbeitet

Ein Problem, auf das man eventuell stößt, besteht darin, daß viel ISPs nur Unterstützung für die von ihnen gelieferten PPP Software bieten. Und diese läuft fast immer nur unter Windows. Vielen ISPs ist Linux als Betriebssystem auf Kundenseite völlig unbekannt, so daß man auf sich selbst gestellt ist. Man sollte also darauf vorbereitet sein, von dieser Seite keine Unterstützung zu finden. Einige Provider verwenden außerdem proprietäre Protokolle, die nur mit spezieller Software für Windows benutzt werden können, wie das z.B. bei AOL der Fall ist.

23.1 Ich kann niemand außer dem PPP Server erreichen.

Gut, die PPP Verbindung scheint also zu laufen und man kann den PPP Server mit seiner IP Nummer (die zweite oder `remote` IP Nummer, die `ifconfig ppp0` anzeigt) anpingen, aber man kann keinen anderen Rechner erreichen.

Als erstes sollte man versuchen, die IP Nummer des Nameservers anzupingen, die man in `/etc/resolv.conf` angegeben hat. Wenn dieses funktioniert, kann man auch andere Rechner als den PPP Server ansprechen, außer der Nameserver hat die gleiche IP Nummer wie der PPP Server. Nun sollte man versuchen, den ISP mit seinem kompletten Internet Namen anzupingen, also z.B.:

```
ping mein.provider.net.au
```

Falls diese *nicht* funktionieren sollte, hat man eine Problem mit der Auflösung des Names in eine IP Nummer. Dieses könnte an einem Tipfehler in der `/etc/resolv.conf` Datei liegen, die ja den Nameserver festlegt, der verwendet werden soll. Man sollte die Adresse in der Datei noch einmal genau mit denen vergleichen, die man von seinem ISP bekommen hat. Falls alles korrekt aussieht, sollte man noch einmal den Support des ISP anrufen und sich vergewissern, daß man die IP Nummern der Name Server korrekt notiert hat.

Falls auch das nicht helfen sollte und der ISP versichert, daß seine Nameserver laufen, liegt das Problem irgendwo anders. In diesem Fall sollte man die eigene Linux Installation noch einmal genau überprüfen und hierbei besonders auf die Zugriffsrechte der Dateien achten.

Falls man die Nameserver des ISPs nicht mittels ihrer IP Nummer anpingen kann, laufen die Nameserver entweder nicht oder es gibt beim ISP ein Routing Problem. In diesem Fall sollte man beim Support des ISPs anrufen.

Ein Möglichkeit ist z.B., daß der ISP einen Linux PPP Server verwendet und dieser vergessen hat, die IP forwarding Option im Kernel einzuschalten.

Ein guter Test ist es, zu versuchen, mit der Software von Microsoft Windows eine PPP Verbindung zu dem Provider aufzubauen. Falls man mit einem anderen Betriebssystem mit demselben Account eine PPP Verbindung aufbauen kann, dann liegt das Problem bei dem eigenen Linux System und *nicht* bei dem ISP.

23.2 Ich kann email versenden, aber keine empfangen.

Wenn man dynamische IP Nummern verwendet, ist dieses normal.

24 Die Benutzung von Internet Diensten mit dynamischen IP Nummern

Wenn man dynamische IP Nummern verwendet und das ist bei den meisten ISPs der Fall, da diese statische IP Nummern nur gegen ein erheblich höheres Entgeld vergeben, muß man die Einschränkungen erkennen, die dadurch entstehen.

Bei der Verwendung von dynamischen IP Nummern macht die eigene Benutzung von angebotenen Diensten keine Probleme. Man kann also ohne Probleme emails mit `sendmail` verschicken, Dateien von Servern mittels `ftp` beziehen, Benutzer auf anderen Rechnern fingern, im Web surfen usw.

Allerdings ist der eigene Rechner *nicht* 24 Stunden am Tag mit dem Internet verbunden und er hat nicht jedesmal die selbe IP Nummer, wenn er mit dem Internet verbunden ist. Aus diesem Grund ist es unmöglich, daß jemand an den Rechner direkt emails schickt oder daß man einen Web oder FTP Server einrichtet.

Wenn man einen WWW oder anderen Server einrichtet, ist dieser für alle Benutzer des Internets solange unbekannt, bis diese wissen, daß dieser gerade mit dem Internet verbunden ist und seine jetzige IP Nummer in Erfahrung gebracht haben. Man kann diese Informationen zwar auf diversen Wegen mit anderen Benutzern austauschen, aber für richtige Server ist dieser Weg nicht geeignet.

Für die meisten Internet Benutzer stellt diese Einschränkung aber kein Problem dar, da sie selbst keine Dienste anbieten wollen, sondern nur emails senden und empfangen möchten, was unter Verwendung des Accounts beim ISP möglich ist, Daten von WWW, FTP und anderen Server abrufen möchten.

24.1 Konfiguration von email

Auch mit dynamischen IP Nummern kann man `sendmail` so konfigurieren, daß es lokal erzeugte emails nach dem Verbindungsaufbau verschickt. Die Konfiguration von `sendmail` ist recht schwierig. Aus diesem Grund versuchen wir hier nicht, dieses zu beschreiben. Man sollte `sendmail` so einrichten, daß der ISP als *smart relay* benutzt wird (die `sendmail.cf DS` Option). `sendmail` kontaktiert dann nicht jeweils direkt die Server, die das email erhalten sollen, sondern reicht alle an den ISP Server weiter, der das Verteilen dann übernimmt. So spart man kostbare Zeit. Weitere Informationen zur Konfiguration von `sendmail` findet man in der mitgelieferten Dokumentation und in der *Mail HOWTO*.

Wenn man `sendmail` erst einmal konfiguriert hat, möchte man sicherlich, daß die emails sofort nach dem Verbindungsaufbau automatisch verschickt werden. Um dieses zu erreichen, fügt man folgdende Zeile

```
sendmail -q
```

in das `ip-up` Skript ein.

Der Empfang von emails ist ein Problem mit dynamischen IP Nummern. Die meisten ISP bieten heute kein UUCP mehr an. So bleibt meistens nur noch das POP Protokoll. Man holt hierbei die Mails an den eigenen Account beim ISP mit einem Programm wie `popclient` ab. Allerdings braucht man fuer jeden lokalen Benutzer einen eigenen Account beim Provider.

24.2 Einrichten eines lokalen Nameservers

Während man natürlich den Domain Name Server des ISP benutzten kann, kann man auch einen eigenen lokalen caching only Nameserver einrichten, der vom `ip-up` Skript gestartet wird. Der Vorteil eines lokal laufenden caching only Name Servers liegt darin, daß man Zeit und Bandbreite spart, wenn man häufig die gleichen Adressen während einer Verbindung mit dem ISP anspricht.

Die DNS Konfiguration für einen caching only Name Server, der eine forwaders Zeile in der `named.boot` Datei verwendet, die auf den DNS des ISPs zeigt, ist relativ einfach. Das O'Reilly Buch *DNS and Bind* erklärt alles hierfür notwendige. Außerdem existiert eine *DNS HOWTO*.

Man sollte allerdings aus Gründen der Nettiquette den ISP vor dem Einrichten eines zweite, caching only Nameserver in dem Domain des ISPs diesen um Erlaubnis fragen. Richtig konfiguriert wird der eigene DNS dem ISP keine Probleme machen, wenn er allerdings nicht richtig konfiguriert wird, kann es zu Problemen kommen.