

Rainer Krienke

Kommunikation unter Linux

2., überarbeitete und erweiterte Auflage



Alle in diesem Buch enthaltenen Programme, Darstellungen und Informationen wurden nach bestem Wissen erstellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund ist das in dem vorliegenden Buch enthaltene Programm-Material mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren und die SuSE Linux AG übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials, oder Teilen davon, oder durch Rechtsverletzungen Dritter entsteht.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann verwendet werden dürften.

Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen. Die SuSE Linux AG richtet sich im Wesentlichen nach den Schreibweisen der Hersteller. Andere hier genannte Produkte können Warenzeichen des jeweiligen Herstellers sein.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Druck, Fotokopie, Microfilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.
ISBN 3-935922-63-9

© 2003 SuSE Linux AG, Nürnberg (<http://www.suse.de>)

Umschlaggestaltung: Fritz Design GmbH, Erlangen

Gesamtlektorat: Nicolaus Millin

Fachlektorat: Holger Achtziger, Sven Dummer, Michael Eicks, Franz Hassels, Uwe Hering, Fabian Herschel, Pieter Hollants, Lars Knoke, Reinhard Max, Wolfgang Rosenauer, Henne Vogelsang

Satz: \LaTeX

Druck: Kösel, Kempten

Printed in Germany on acid free paper.

Vorwort

In der heutigen Zeit hat der Begriff „Kommunikation“ einen hohen Stellenwert, gerade in Bezug auf Computer-Systeme, die nicht mehr nur mit der Bewältigung von Rechen- oder Office-Aufgaben betraut werden, sondern zunehmend auch als Kommunikationszentrale dienen. In dieser Funktion werden von den Rechnern neben den „klassischen Diensten“ (Versenden und Empfangen von E-Mail oder das Surfen im Web) auch solche angeboten, mit deren Hilfe sie beispielsweise als Telefonanlage und Anrufbeantworter eingesetzt werden können.

Dieses Buch bietet dem Leser einen Einstieg in die Möglichkeiten, die genannten und weitere Kommunikationsformen unter dem wohl populärsten Unix-Vertreter – Linux – zu nutzen. Unix ist schon seit Beginn seiner Entwicklung in den 1970er Jahren dafür bekannt, daß es einerseits außerordentlich stabil, andererseits bei der Vernetzung mit anderen Systemen sehr kommunikationsfreudig ist. Linux baut auf diesen Eigenschaften auf und bietet insbesondere für neuere Bereiche, wie z. B. Multimedia, Unterstützung, die bisher eher anderen Systemen vorbehalten war.

Die in diesem Buch getroffene Themen-Auswahl soll den Leser zum einen in die Lage versetzen, die klassischen Dienste komfortabel und effizient zu nutzen. Hierzu werden Werkzeuge vorgestellt, mit denen z. B. das offline Browsen im World Wide Web möglich ist, wie auch Methoden zur Automatisierung von Routineaufgaben, wie z. B. die automatische Abfrage neuer Mails oder Newsartikel beim Verbindungsaufbau zum Internet-Provider. Weitere Abschnitte beschreiben nützliche Werkzeuge z. B. zum (automatischen) Download von Dateien. Für Anbieter von Internet-Diensten werden der Aufbau eines PPP- und eines FTP-Servers sowie die Erstellung von Web-Seiten mit dynamischem Inhalt beschrieben.

Neben diesen Internet-spezifischen Themen wird der Leser ausführlich in die Nutzung weiterer leistungsfähiger Programmpakete, wie z. B. für das Fax-Versenden und den Fax-Empfang mit Hilfe eines Modems oder einer ISDN-Karte, eingeführt. Weitere Abschnitte beschreiben die Realisierung eines Anrufbeantworters auf Basis eines Modems oder einer ISDN-Karte sowie die Nutzung von Multimediadiensten, wie Fernsehen und Radio unter Linux. Auch die Themen Rechnernetzung über die parallele Schnittstelle, die drahtlose Kommunikation über Infrarot als auch der Aufbau eines Wireless LANs unter Linux werden behandelt.

Die Darstellung erfolgt stets, indem zunächst Grundlagen beschrieben werden, die dem Leser einen allgemeinen Überblick und wichtige Hintergrundinformationen über das entsprechende Themengebiet bieten, bevor im nächsten Schritt konkrete Lösungen praxisnah und ausführlich vorgestellt werden. Auf diese Weise gewinnt der Leser Einblick in die Zusammenhänge und erhält darüber hinaus

die Möglichkeit, die gewünschte Lösung rasch und seinen Wünschen entsprechend zu konfigurieren.

Das Buch ist nach der Art der Kommunikation in Kapitel zur Datenkommunikation, zu Internet-Diensten, zur Audio- und Videokommunikation sowie zum Komplex Vernetzung klar gegliedert. Grundlegende Konfigurationsschritte, die als Basis für die Nutzung der genannten Kommunikationsformen dienen, sind im 2. Kapitel (*Grundkonfiguration*) zusammengefaßt. Der Interessierte kann das Buch kapitelweise von Anfang bis Ende durchlesen oder auch einzelne für sich relevante Kapitel direkt auswählen. Um die enge Beziehung der einzelnen Themenbereiche deutlich zu machen, enthalten die einzelnen Abschnitte – wo immer sinnvoll – Querverweise sowie zahlreiche Hinweise auf weiterführende Informationen im World Wide Web. So wird der Leser in die Lage versetzt, jederzeit die aktuellsten Entwicklungen auf den verschiedenen Gebieten zu verfolgen. Ein Glossar rundet die Darstellung ab und gibt Gelegenheit zum raschen Nachschlagen der im Buch erwähnten Fachbegriffe.

Die Entwicklung auf dem Gebiet der Kommunikation schreitet stetig voran, wodurch dem Benutzer immer neue Möglichkeiten eröffnet werden bzw. immer mehr Hardware (wie z. B. Multimedia-Karten) genutzt werden kann. Die Darstellung in diesem Buch beschreibt den zum Zeitpunkt der Manuskript-Erstellung aktuellen Stand. Durch die Weiterentwicklung von Anwendungssoftware und Gerätetreibern kann nicht ausgeschlossen werden, daß einzelne Beschreibungen nicht mehr in allen Details mit den Eigenarten neuerer Software übereinstimmen. In diesem Fall kann der Leser auf die Dokumentation der entsprechenden Software zurückgreifen. Darüber hinaus stehen aktuelle Informationen zu diesem Buch unter der URL

<http://www.uni-koblenz.de/~krienke/books/linuxkom.html>

zur Verfügung. Über diese URL kann auch am besten auf die im Buch abgedruckten URLs zugegriffen werden. Schließlich können an diese Adresse auch Kritik sowie Verbesserungsvorschläge zum Buch gesendet werden.

An dieser Stelle möchte ich allen, die mich bei der Erstellung des Buchs unterstützt haben, danken. Mein besonderer Dank geht an meine Frau, die mir im Laufe der Zeit zwar nicht mehr geglaubt hat, daß ich mit ihr, und nicht mit dem Rechner verheiratet bin, mich aber dennoch durch die Übernahme vieler Aufgaben beim Schreiben unterstützt hat. Mein weiterer Dank gilt Herrn Nicolaus Millin, dem ich durch seine Arbeit bei SuSE PRESS die Idee und die Entstehung dieses Buchs verdanke. Darüber hinaus möchte ich Frau Dr. Sonja Weyher, Herrn Ralf Bremecke sowie Herrn Reinhard Max, Herrn Holger Achtziger und Herrn Lars Knoke für ihre konstruktive Kritik danken.

Koblenz, Juni 2000

Rainer Krienke

Vorwort zur 2. Auflage

Die Entwicklung von Linux verläuft immer noch sehr rasant. Um der wachsenden Zahl von Anwendungen und Techniken auf dem Sektor der Kommunikation unter Linux gerecht zu werden, bietet diese Neuauflage zum einen ein Update für die schon in der ersten Auflage beschriebenen Themengebiete, um den aktuellen Stand der Technik widerzuspiegeln. Darüber hinaus wurden bestehende Themen ausgebaut und neue Themen aufgegriffen.

Zu den wesentlichen Erweiterungen gehören z. B. die in dieser Auflage stark erweiterte Beschreibung von T-DSL und der Nutzungsmöglichkeiten zum Fax-Versand und -Empfang über ISDN. Im Bereich der Internet-Kommunikation werden weitere nützliche Werkzeuge vorgestellt.

Das Kapitel Audiokommunikation wurde um eine detaillierte Beschreibung ergänzt, die zeigt, wie man eigene Audio-CDs erstellt und Musiktitel zu MP3 oder Ogg-Vorbis Dateien „rippt“. Auf diese Weise können selbst zusammengestellte CDs mit vielen Stunden Musik hergestellt werden.

Das Kapitel Videokommunikation enthält jetzt eine ausführliche Beschreibung der Möglichkeiten, IP-Telefonie und Videokonferenzen unter Linux zu nutzen. Ebenfalls neu ist die Beschreibung der Thematik von Aufnahme und Wiedergabe von Videosequenzen (Videorecorder und Video-Player) als auch von Streaming von Videodaten.

Schließlich enthält das Kapitel über Vernetzungsmöglichkeiten mit dem neuen Abschnitt über Wireless LANs, also drahtloser Funknetzwerke, die Beschreibung der Nutzungsmöglichkeiten einer Technologie unter Linux, die in sehr kurzer Zeit bereits eine sehr große Verbreitung gefunden hat. Wireless LANs werden sowohl in Firmen als auch im Home-Bereich genutzt und finden darüber hinaus immer größere Verbreitung an öffentlichen Plätzen, wie z. B. in Internet Cafes oder Flughäfen. Sie ermöglichen den drahtlosen Zugriff, z. B. auf Daten im Internet von einem Laptop aus.

Koblenz, Februar 2003

Rainer Krienke

Inhaltsverzeichnis

1	Einleitung	1
1.1	Was das Buch bietet	1
1.2	Was das Buch nicht ist	3
1.3	Schreibkonventionen	3
2	Grundkonfiguration	5
2.1	Konfiguration analoger Modems	5
2.1.1	Die serielle Schnittstelle	6
2.1.1.1	Etwas Theorie zur Funktionsweise	6
2.1.1.2	Übertragungsarten	7
2.1.1.3	Verkabelung	8
2.1.1.4	Datenflußprotokolle	10
2.1.2	Anschluß eines Modems	12
2.1.3	Modem-Protokolle	13
2.1.3.1	Modulationsverfahren und Standards	14
2.1.3.2	Standards zur Fehlerkorrektur	15
2.1.3.3	Standards zur Datenkompression	15
2.1.4	Der AT-Kommandosatz	16
2.1.5	Konfiguration für die Einwahl beim ISP über ein Modem	19
2.1.5.1	Konfiguration mit Bordmitteln	20
2.1.5.2	Einwahlkonfiguration mit <code>wvdial</code>	20
2.1.5.3	Einwahlkonfiguration mit <code>yast2</code>	26
2.1.5.4	Einwahlkonfiguration mit <code>kppp</code>	29

2.2	Konfiguration von ISDN-Karten	36
2.2.1	ISDN-Grundlagen	37
2.2.2	ISDN unter Linux	40
2.2.3	I4L basierte Nutzung von ISDN	40
2.2.3.1	Lowlevel-Konfiguration einer Karte	40
2.2.3.2	Einwahlkonfiguration mit <code>yast2</code>	44
2.2.3.3	Manuelle ISDN-Einwahl mit I4L-Tools	46
2.2.3.4	I4L-Utilities	48
2.2.4	CAPI basierte Nutzung von ISDN	48
2.2.4.1	Installation der AVM-CAPI	49
2.2.4.2	Konfiguration der I4L-Kompatibilität	61
2.2.4.3	Einwahlkonfiguration mit der AVM CAPI	62
2.3	ADSL unter Linux	68
2.3.1	Grundlagen	69
2.3.2	Einführung in T-DSL	71
2.3.3	T-DSL-Konfiguration mit <code>pppoe</code> -Support im Kernel	73
2.3.3.1	Konfiguration mit <code>yast2</code>	75
2.3.3.2	Manuelle Konfiguration	80
2.3.4	DSL-Konfiguration mit RP-PPPoE	88
2.3.4.1	Skript-basierte RP-PPPoE-Konfiguration	90
2.3.4.2	Graphische Konfiguration	97
2.3.5	DSL-Kostenkontrolle	99
2.3.6	Das MTU-Problem	101
2.4	SuSE Linux-Werkzeuge zur Internet Einwahl	104
2.4.1	Funktionsprinzip	106
2.4.2	Konfiguration	107
2.5	Wichtige Informationsquellen im WWW	109
3	Datenkommunikation	111
3.1	Faxen mit Modems	111
3.1.1	Grundlagen	111
3.1.2	Modems, Protokolle und Standards	113
3.1.2.1	Fax-Gruppen	113

3.1.2.2	Fax-Klassen	115
3.1.3	Fax on demand und Fax-Polling	118
3.1.4	HylaFAX	119
3.1.4.1	Funktionsweise von HylaFAX	121
3.1.4.2	Installation und Grundkonfiguration	123
3.1.5	Weiterführende Server-Konfiguration	141
3.1.5.1	Senden, Verwalten und Ansehen eines Fax	151
3.1.5.2	Deckblätter	173
3.1.6	Mgetty und Sendfax	177
3.1.6.1	Grundkonfiguration	178
3.1.6.2	Empfang eines Fax	184
3.1.6.3	Konvertieren und Versenden eines Fax mit Hilfe von sendfax	185
3.1.6.4	Komfortabler Fax-Versand mit faxspool	187
3.1.6.5	Fax-Versand mit graphischem Frontend	190
3.2	Faxen über ISDN	191
3.2.1	Faxversand über die AVM-CAPI-Lösung	192
3.2.1.1	Installation und Konfiguration	193
3.2.1.2	Senden und Empfangen eines Fax	201
3.2.1.3	Betrieb eines Faxservice mit capi4hylafax am Anlagenanschluß	206
3.2.2	Weitere Faxlösungen für die AVM B1-Karte	212
3.2.2.1	Kernel-Konfiguration	213
3.2.2.2	B1-Basiskonfiguration	216
3.2.2.3	Versenden und Empfangen eines Fax	218
3.3	Fax-Versand über einen Fax-Drucker	220
3.3.1	Der Fax-Drucker unter KDE	220
3.3.2	Das faxprint-Paket für SuSEFax	222
3.3.3	Das faxprt-Paket	223
3.4	Scannen von Dokumenten für den Fax-Versand	223
3.4.1	Installation und Konfiguration	224
3.4.2	Werkzeuge zum Scannen	225
3.4.3	Scannen und Faxen	226

3.5	Fax-Versand über das Internet	227
3.6	Nützliche Informationsquellen im WWW	229
3.7	Direkte serielle Kommunikation	230
3.7.1	Grundlagen	230
3.7.2	minicom	231
3.7.2.1	Grundkonfiguration	231
3.7.2.2	Arbeiten mit minicom	234
3.7.2.3	Dateitransfer unter minicom	237
4	Internet-Dienste	241
4.1	Werkzeuge zur Nutzung grundlegender Internet-Dienste	241
4.1.1	Terminal-basiertes Web-Browsen mit lynx	241
4.1.1.1	Installation und Konfiguration	242
4.1.1.2	Navigation	244
4.1.2	File transfer mit ncftp	247
4.1.2.1	Ablauf einer FTP-Sitzung	248
4.1.2.2	Das FTP-Werkzeug ncftp	248
4.1.2.3	Modi	249
4.1.2.4	Visual und Line Mode	249
4.1.2.5	Der Colon Mode	256
4.1.2.6	Kommandozeilen-Optionen	257
4.1.2.7	Wesentliche Merkmale von ncftp (Version 3)	258
4.1.3	File transfer mit wget	259
4.1.3.1	Rekursiver Download	261
4.1.3.2	Authentifizierung	264
4.1.3.3	Die wget-Konfigurationsdateien	264
4.1.3.4	Zusammenfassung der wichtigsten Optionen	265
4.2	Content-Web-Seiten mit NewsClipper	268
4.2.1	Die Funktionsweise von NewsClipper	268
4.2.1.1	Grundlagen	269
4.2.1.2	Dokumentation	269
4.2.1.3	Ein Beispiel	269
4.2.1.4	Handler	272

4.2.2	Installation und Konfiguration	273
4.2.2.1	Installation der NewsClipper Software	274
4.2.2.2	Installation der benötigten Perl-Module	277
4.2.2.3	Download und Installation der Handler	279
4.2.3	Erstellen einer Content-Web-Seite	279
4.2.4	Erstellen eigener Handler	280
4.3	Offline im WWW browsen mit <code>wwwoffline</code>	290
4.3.1	Arbeitsweise	291
4.3.2	Installation und Grundkonfiguration	296
4.3.3	Weitere Konfigurationsmöglichkeiten	299
4.3.3.1	<code>wwwoffline</code> -Konfigurationsoptionen	300
4.3.3.2	Automatische Proxy-Browser-Konfiguration	303
4.3.3.3	Passwortschutz der Konfigurationsdatei	307
4.3.3.4	Automatisches Setzen des <code>wwwoffline</code> -Modus	307
4.3.3.5	Nutzung von <code>wwwoffline</code> mit mehreren Providern	308
4.3.3.6	<code>wwwoffline</code> in kleinen Netzwerken	309
4.3.4	Das <code>wwwoffline</code> -DNS-Problem	312
4.3.5	Verwaltung und Überwachung	314
4.4	Offline News lesen mit <code>leafnode</code>	316
4.4.1	Arbeitsweise	317
4.4.2	Installation und Grundkonfiguration	318
4.4.3	Weitere Konfigurationsmöglichkeiten	321
4.4.4	Administration von <code>leafnode</code>	323
4.4.4.1	Automatisiert neue Artikel holen	323
4.4.4.2	Spezielle Administrationsoperationen	324
4.5	Offline Mail lesen mit <code>fetchmail</code>	325
4.5.1	Arbeitsweise	326
4.5.2	Installation und Konfiguration	327
4.5.3	Wichtige Konfigurationsoptionen	330
4.5.3.1	Globale Optionen	330
4.5.3.2	Mail-Server-Optionen	332
4.5.3.3	Benutzer-Optionen	333

4.5.4	Spezielle Konfigurationsmöglichkeiten	335
4.5.4.1	Sichere Übertragung von Mail über <code>ssh</code>	335
4.5.4.2	Mail von verschiedenen Providern abrufen	336
4.5.4.3	Multidrop-Konfiguration	338
4.6	Mail versenden und lesen mit <code>masqmail</code>	339
4.6.1	Arbeitsweise	341
4.6.2	Installation	342
4.6.3	Grundkonfiguration	342
4.6.3.1	Ermittlung des Verbindungszustands	348
4.6.4	Mail-Routing Konfiguration	349
4.6.5	Arbeiten mit <code>masqmail</code>	354
4.6.6	Abrufen von Mail mit <code>masqmail</code>	356
4.7	Aufbau eines PPP-Servers	359
4.7.1	Voraussetzungen	360
4.7.2	Konfigurationsdateien	362
4.7.3	Ein einfacher PPP-Client	364
4.7.4	Authentifikationsprotokolle	368
4.7.4.1	Authentifizierung mit PAP	369
4.7.4.2	Authentifizierung mit CHAP	371
4.7.5	Serverkonfiguration	373
4.7.5.1	PPP-Server mit analogem Modem-Zugang	373
4.7.5.2	PPP-Server, der auf der AVM-CAPI basiert	377
4.7.5.3	Weitere Möglichkeiten der Nutzung von PPP unter ISDN	382
4.7.6	Häufig verwendete Konfigurationsoptionen	384
4.7.7	Skripten	387
4.7.8	Routing	393
4.7.9	Rechner-Rechner-Verbindung mit <code>pppd</code>	395
4.8	Einrichten eines FTP-Servers	397
4.8.0.1	Funktionsweise	398
4.8.0.2	Installation und Konfiguration	401
4.8.0.3	Die <code>ftpaccess</code> -Datei	404
4.8.0.4	Die Datei <code>ftpusers</code>	412

4.8.0.5	Die Datei <code>ftpconversions</code>	412
4.9	Wichtige Informationsquellen im WWW	413
5	Audiokommunikation	415
5.1	Grundlagen	415
5.1.1	Digitalisierung von Audiodaten	415
5.1.1.1	Digitalisierung	416
5.1.2	Komprimierung von Audiodaten	417
5.1.2.1	A-Law und U-Law	417
5.1.2.2	ADPCM	417
5.1.2.3	GSM	418
5.1.2.4	MPEG	418
5.1.2.5	Ogg Vorbis	418
5.1.3	Wichtige Audio-Dateiformate	418
5.1.3.1	Selbstbeschreibende Audio-Dateiformate	419
5.1.3.2	Audio-Dateiformate ohne Meta-Information	420
5.1.4	Bearbeiten/Konvertieren von Audiodaten	421
5.1.4.1	Soundkonvertierung mit <code>sox</code>	421
5.1.4.2	Die <code>pvf</code> -Tools	425
5.1.4.3	Die <code>vbox</code> -Tools	427
5.2	Abspielen und Aufnahmen von Audiodaten	428
5.2.1	Kommandozeilenorientierte Werkzeuge zum Aufnehmen und Abspielen	429
5.2.1.1	<code>play</code> und <code>rec</code>	429
5.2.1.2	Die Programme <code>wavplay</code> und <code>wavrec</code>	430
5.2.1.3	Das Programm <code>vboxplay</code>	431
5.2.1.4	Der MP3-Player <code>mpg123</code>	431
5.2.1.5	Der Ogg-Vorbis-Player <code>ogg123</code>	432
5.2.2	X Window-basierte Werkzeuge	432
5.2.2.1	Das Programm <code>noatun</code>	432
5.2.2.2	Das Programm <code>xmms</code>	433
5.2.2.3	Das Programm <code>krecord</code>	433
5.3	Erstellen und Bearbeiten von Audio-CDs	434

5.3.1	Basis-Werkzeuge	434
5.3.1.1	Utilities zum Auslesen von Audio-CDs	434
5.3.1.2	Encoder-Programme	435
5.3.1.3	Tools zum Schreiben von CDs	436
5.3.2	Rippen von Audio-CDs	437
5.3.2.1	Rippen mit dem <code>konqueror</code>	437
5.3.2.2	Rippen mit <code>grip</code>	439
5.3.3	<code>ripit</code>	441
5.3.4	Zusammenstellung eigener Audio-CDs	444
5.3.4.1	Erstellen einer Audio-CD mit Hilfe von <code>cdrecord</code> und <code>mkisofs</code>	445
5.3.4.2	Erstellen einer Audio-CD mit <code>koncd</code>	450
5.3.4.3	Erstellen einer Daten-CD mit <code>cdbakeoven</code>	452
5.4	Linux als Anrufbeantworter	454
5.4.1	Anrufe beantworten mit dem <code>vgetty</code> -Paket	454
5.4.1.1	Installation	455
5.4.1.2	Manuelle Konfiguration	455
5.4.1.3	Konfiguration mit <code>kvoice</code>	465
5.4.1.4	Log-Meldungen	467
5.4.1.5	Fernabfrage mit DTMF-Ton-Auswertung	467
5.4.2	Das <code>vbox</code> -Paket	470
5.4.2.1	Installation	470
5.4.2.2	Manuelle Konfiguration	471
5.4.2.3	Konfiguration mit <code>SuSEVbox</code>	489
5.4.2.4	Steuerung des Anrufbeantworters	490
5.4.2.5	Fernabfrage mit <code>raccess4vbox</code>	492
5.4.3	Nützliche Informationsquellen im WWW	499
6	Videokommunikation	501
6.1	Das "Video for Linux"-Projekt	502
6.2	Hardware	503
6.2.1	Unterstützte Hardware	503
6.2.2	Automatische Konfiguration einer TV-Karte	504

6.2.3	Konfiguration einer BTTV-TV/Radio-Karte	505
6.2.3.1	Download des BTTV-Treibers	505
6.2.3.2	Überprüfen der Konfiguration der Kernel-Treiber- Module	506
6.2.3.3	Einrichten der Gerätedateien	506
6.2.3.4	Laden der Treiber	507
6.2.3.5	Anpassen der Datei <code>modules.conf</code>	514
6.3	TV-Applikationen	515
6.3.1	Einführung in die Fernsehtechnik	515
6.3.2	Fernsehen mit <code>xawtv</code>	517
6.3.3	Audio-Aufzeichnungen mit <code>xawtv</code>	522
6.3.4	Weitere Möglichkeiten	522
6.3.5	Fernsehen mit <code>kwintv</code>	523
6.3.6	Fernbedienung von <code>xawtv</code> und <code>kwintv</code>	528
6.3.6.1	Manuelle LIRC-Installation	528
6.3.6.2	LIRC-Konfiguration	532
6.3.6.3	Start des <code>lircd</code> -Prozesses	532
6.3.6.4	IR-Konfiguration für <code>xawtv</code>	536
6.3.6.5	IR-Konfiguration für <code>kwintv</code>	537
6.3.7	Weitere TV-Applikationen	539
6.4	Radio-Applikationen	540
6.4.1	Radio hören mit den <code>fmtools</code>	540
6.4.2	Radio hören mit <code>kradio</code>	541
6.4.3	Die <code>kderadio</code> -Applikation	542
6.5	Darstellung von Videotext	542
6.5.1	Videotext lesen mit <code>alevt</code>	543
6.5.2	Videotext und Inter casting mit <code>vbidecode</code>	544
6.5.2.1	<code>vbidecode</code> -Konfiguration	545
6.5.2.2	Betrachten von <code>vbidecode</code> -Dateien	546
6.6	Weitere Software	548
6.6.1	Werkzeuge zum Videoschnitt	548
6.6.2	Nützliche Informationsquellen im WWW	548
6.6.2.1	Grundlegende Informationen	548

6.6.2.2	Informationen zu ausgewählten Multimedia-Programmen	549
6.6.2.3	Technische Informationen	549
6.7	Verwendung von Kameras unter Linux	550
6.7.1	Grundlagen	550
6.7.1.1	Kameraanschluß an TV-Karte	551
6.7.1.2	USB-basierte Kameras	551
6.7.1.3	Kameras zum Anschluß an den Parallelport	552
6.7.1.4	Netzwerkbasierte Web-Kameras	552
6.7.2	Anschluß und Nutzung einer USB-basierten Kamera	552
6.7.2.1	Installation und Laden von Treibern	553
6.7.2.2	Anwender-Software	556
6.7.3	Nützliche Informationsquellen in WWW	561
6.8	Aufnahme und Wiedergabe von Videosequenzen	562
6.8.1	Übersicht über wichtige Codecs und Dateiformate	562
6.8.1.1	AVI	563
6.8.1.2	RealMedia	564
6.8.1.3	MPEG-1,2,4	564
6.8.1.4	DivX	565
6.8.1.5	RGB	565
6.8.1.6	YUV	566
6.8.1.7	Motion JPEG	567
6.8.2	Aufnahme von Fernsehsendungen mit dvr	567
6.8.3	Videoerstellung und Videostreaming mit ffmpeg	569
6.8.3.1	Aufzeichnen von Videomaterial	571
6.8.3.2	Video-Streaming mit ffserver	572
6.8.4	Wiedergabe digitaler Videodaten	576
6.8.4.1	Der Media-Player MPlayer	576
6.8.4.2	noatun	581
6.8.4.3	xine	581
6.8.4.4	Ogle	584
6.8.4.5	aktion und xanim	585
6.8.4.6	XMoview	585

6.8.5	Nützliche Informationsquellen im WWW	585
6.8.5.1	Video-Dateiformate und -Codecs	586
6.8.5.2	Video-Player und Recorder	587
6.9	Video-Konferenzen und IP-Telefonie	587
6.9.1	Übersicht über verschiedene Systeme	588
6.9.2	Protokolle und Codecs	589
6.9.2.1	Das H.323-Protokoll	590
6.9.2.2	Weitere bekannte Protokolle	595
6.9.2.3	Das RTP-Benutzer-Protokoll	595
6.9.2.4	Verwendete Codecs	596
6.9.3	Technische Voraussetzungen für die IP-Telefonie	598
6.9.4	Software für H.323-basierte Telefonie unter Linux	598
6.9.4.1	Videokonferenzen mit <code>gnomemeeting</code>	599
6.9.4.2	Die Anwendung <code>ohphone</code>	609
6.9.4.3	Konferenzen mit mehr als zwei Teilnehmern	613
6.9.5	Interessante Informationen aus dem World Wide Web	617
6.9.5.1	Links zu Anwendungen	617
6.9.5.2	Links zum H.323-Protokoll	618
6.9.5.3	Informationen zu Codecs	619
7	Rechnervernetzung ohne Netzkabel	621
7.1	Vernetzung mit PLIP	622
7.1.1	Grundkonfiguration	622
7.1.2	Konfiguration des Interface	625
7.1.3	Das Lap-Link-Kabel	627
7.2	Vernetzung mit IRDA	628
7.2.1	Grundlagen	628
7.2.2	Protokolle	629
7.2.3	Installation und Konfiguration	631
7.2.4	Nutzung verschiedener IrDA-Dienste	640
7.2.5	Nützliche Informationsquellen im WWW	642
7.3	Wireless LANs unter Linux	643
7.3.1	Einführung	643

7.3.1.1	Technische Grundlagen	644
7.3.1.2	Wireless und Linux	645
7.3.2	Struktur eines WLANs	647
7.3.2.1	Hinweise zur Standortwahl eines Access Points	649
7.3.2.2	Aufbau und Konfiguration eines APs	650
7.3.3	Konfiguration von WLAN-Karten	654
7.3.3.1	Treibervarianten	654
7.3.3.2	Inbetriebnahme von PCI-WLAN-Karten	655
7.3.3.3	Inbetriebnahme von PCMCIA-WLAN-Karten	656
7.3.3.4	Wireless-Konfiguration	660
7.3.3.5	Nutzung der WLAN-NG-Treiber	673
7.3.4	Sicherheit in WLANs	674
7.3.4.1	Die SSID eines WLANs	675
7.3.4.2	MAC-basierte Autorisierung am Access-Point	676
7.3.4.3	Verschlüsselte Daten-Übertragung mit WEP	678
7.3.4.4	Port und Protokoll-Filterung am Access-Point	683
7.3.4.5	Schutz des kabelgebundenen Netzwerks	683
7.3.4.6	VPNs	685
7.3.5	Linux-WLAN-Tools	686
7.3.5.1	kwifimanager	686
7.3.5.2	Administrations-Tools	687
7.3.6	Schwierigkeiten mit PCMCIA-Karten	689
7.3.7	Nützliche Informationsquellen im WWW	693

Kapitel 1

Einleitung

Linux ist ein überaus vielseitiges Betriebssystem. Neben seinen schon allgemein bekannten Stärken (z. B. Effizienz und Stabilität), die die Verbreitung von Linux insbesondere im Server-Bereich weiter zunehmen lassen, bietet Linux auch für den Desktop-Anwender viele interessante Möglichkeiten. Hierzu gehört insbesondere die Fähigkeit zur Kommunikation mit anderen Systemen, deren interessanteste Formen in diesem Buch dargestellt werden. Neben der Beschreibung von Konfiguration und Anwendung der entsprechenden Software wird dem Leser in einem einleitenden Teil zuvor stets ein Überblick über das jeweilige Thema gegeben.

1.1 Was das Buch bietet

Das Buch gliedert sich in die folgenden Bereiche bzw. Kapitel:

- Grundkonfiguration (Kapitel 2)
- Datenkommunikation (Kapitel 3)
- Internet-Dienste (Kapitel 4)
- Audiokommunikation (Kapitel 5)
- Videokommunikation (Kapitel 6)
- Rechnernetzwerk ohne Netzkabel (Kapitel 7)

Kapitel 2 führt den Leser in die grundlegende Konfiguration der verschiedenen Kommunikationsbasisdienste ein. Hier wird z. B. die Anbindung und Konfiguration von Modems zur Einwahl bei einem Provider beschrieben. Neben Modems wird auch ausführlich auf die Konfiguration von ISDN und dem High-Speed Netzwerkzugang über T-DSL eingegangen. Kapitel 2 stellt somit die Basis zur Nutzung vieler der in den weiteren Kapiteln beschriebenen Dienste dar.

Kapitel 3 und 4 behandeln alle Dienste, bei denen Daten im engeren Sinn¹ übertragen werden. Hierzu zählt natürlich die Anbindung eines Rechners an einen Internet Service Provider (ISP). Es werden interessante Möglichkeiten beschrieben, die es erlauben, die Zugangszeit und damit auch die Kosten für einen Rechner oder auch ein kleines Netzwerk zu minimieren, indem offline im World Wide Web recherchiert und ebenfalls offline Mail und News aus dem →USENET gelesen werden kann. Auch auf den Aufbau eines eigenen Einwahl-Servers (PPP-Server) oder eines FTP-Servers wird ausführlich eingegangen. In diesem Rahmen wird auch beschrieben, wie einfach unter Linux eine Vernetzung von zwei Rechnern sein kann, um z. B. zwischen zwei Laptops Daten auszutauschen. Ein weiterer Punkt, der in den Kapiteln *Datenkommunikation* bzw. *Internet-Dienste* behandelt wird, sind Fax-Versand und -Empfang. Für Linux existieren sehr leistungsfähige Fax-Systeme, die auf Basis eines Modems oder auch der ISDN-CAPI arbeiten können, vollkommen frei verfügbar sind und sich keineswegs hinter kommerziellen Lösungen verstecken müssen. Neben der Beschreibung der Konfiguration des Fax-Servers werden auch Anwender-Programme (Frontends) beschrieben, die es dem Benutzer gestatten, auf einfache Weise Faxe zu versenden.

Das Kapitel *Audiokommunikation* behandelt Dienste, bei denen Audio-Daten verarbeitet werden. Neben einer Einführung in die Digitalisierung und Speicherung dieser Daten werden verschiedene Software-Lösungen vorgeführt, mit deren Hilfe ein Linux-Rechner als Anrufbeantwortersystem (inkl. solcher Funktionen wie z. B. Fernabfrage) sowohl für analoge Telefonanschlüsse als auch für einen ISDN-Anschluß eingerichtet werden kann. Ein weiteres in diesem Kapitel dargestelltes Thema betrifft das Erstellen eigener Audio-CDs sowie das „Rippen“ von Musikstücken, also das Erstellen von MP3- bzw. Ogg Vorbis-Dateien unter Linux.

In dem folgenden Kapitel, *Videokommunikation*, wird der Betrieb von Multimedia-Karten, wie z. B. einer TV-Karte, beschrieben. Darüber hinaus wird auf Anwendungen zum Fernseh- und Radioempfang wie auch zur Nutzung der Videotext- und Interacting-Angebote eingegangen. Ein weiterer Punkt dieses Kapitels ist die Beschreibung von Videoplayern, mit denen z. B. Filme abgespielt werden können, die in Standard-Formaten, wie beispielsweise MPEG-2, kodiert sind. Ein weiterer Abschnitt dieses Kapitel führt den Leser in die Möglichkeiten der IP-Telefonie und Video-Konferenzen unter Linux ein. Neben der Darstellung der zugrunde liegenden Konzepte werden die wichtigsten Linux-Anwendungen für die IP-Telefonie und für die Durchführung von Videokonferenzen vorgestellt.

Abgeschlossen wird das Buch mit einer Beschreibung von Möglichkeiten, Rechner ohne Verwendung einer Netzwerkkarte miteinander zu vernetzen. Der interessanteste Punkt ist hier sicherlich die Beschreibung der funkbasierten Vernetzung über ein Wireless LAN nach dem 802.11b-Standard. Darüber hinaus werden

¹Auch bei Audio- oder Videodiensten werden letztlich „Daten“ übertragen und verarbeitet, die in diesem Buch jedoch getrennt beschrieben werden.

auch weniger bekannte Verfahren, wie z. B. das PLIP-Protokoll beschrieben, bei dem die Vernetzung über eine einfache parallele Schnittstelle erfolgt. Eine weitere interessante Möglichkeit, die auch die Kommunikationen zwischen Rechnern mit unterschiedlichen Systemen (z. B. Linux- und Palm III-Rechner) ermöglicht, ist die drahtlose Datenübertragung über eine IrDA-kompatible Infrarotschnittstelle. Das Kapitel *Vernetzung ohne Netzwerkkarte* bietet eine ausführliche Einführung in diese Themenbereiche.

1.2 Was das Buch nicht ist

Nachdem nun umrissen wurde, was dieses Buch dem Leser bietet, soll auch kurz darauf eingegangen werden, was das Buch *nicht* ist:

- ❑ Das Buch ist keine Einführung in Linux/Unix. Zur optimalen Nutzung von Linux sollte der Benutzer über Grundkenntnisse von Unix verfügen oder sich diese mit entsprechender Literatur aneignen.
- ❑ Das Buch ist keine allgemeine Anleitung zur Installation von Linux selbst. Es wird davon ausgegangen, daß Linux bereits installiert wurde und „funktioniert“. Hierzu sollte insbesondere das der verwendeten Linux-Distribution beiliegende Handbuch zu Rate gezogen werden. Darüber hinaus finden sich auch im Internet zahlreiche Informationen, wobei hier insbesondere das Linux Documentation Project (<http://www.linuxdoc.org/>) zu nennen ist.
- ❑ Das Buch behandelt nicht die Grundlagen der Bedienung von Benutzeroberflächen wie z. B. KDE. Darüber hinaus wird auch nicht auf die Nutzung grundlegender Anwendungen eingegangen. Hierzu zählen z. B. Web-Browser wie Netscape oder andere Programme zum Lesen von Mail und USENET-News.

1.3 Schreibkonventionen

Das Buch enthält viele Beispiele, anhand derer der Benutzer die Konfiguration oder den Ablauf einer Anwendung leicht nachvollziehen kann. Zu diesem Zweck werden Konventionen zur Schreibweise eingesetzt, bei denen z. B. Bildschirmausgaben durch eine Grau-Hinterlegung deutlich hervorgehoben werden. Im einzelnen gelten folgende Konventionen:

- ❑ Programmnamen sowie Pfadangaben, URLs etc. werden im Teletype-Zeichensatz gesetzt. Ein Beispiel für eine URL ist `http://www.suse.de`.
- ❑ Ausgaben von Programmen werden im Teletype-Zeichensatz gesetzt, wobei der Hintergrund, wie in folgendem Beispiel dargestellt, grau hinterlegt

wird. Sehr lange Ausgaben werden aus Gründen der Übersichtlichkeit durch . . . abgekürzt, falls sie keine zum Verständnis wichtigen Informationen enthalten. Eingaben des Benutzers werden zur Unterscheidung von den Ausgaben eines Programms **fett** dargestellt:

```
tux@erde:/home/tux > echo $SHELL
/bin/bash
tux@erde:/home/tux > cat /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:
tty:x:5:
...
```

In einigen Fällen, z. B. während der Ausführung eines Konfigurationskriptes, muß vom Benutzer u. U. keine Eingabe gemacht, sondern lediglich der vorgegebene Wert durch Drücken der Eingabetaste bestätigt werden. Um auch diese Form der Eingabe sichtbar zu machen, wird ein entsprechendes Tastensymbol für das Drücken der Eingabetaste durch den Benutzer dargestellt – (↵), das ansonsten nicht explizit hinzugefügt wird:

```
...
Protection mode for session logs [0600]? (↵)
Protection mode for modem [0600]? 0644 (↵)
Rings to wait before answering [1]? 3
Modem speaker volume [off]? (↵)
...
```

- An einigen Stellen des Texts wird statt eines konkreten Werts eine Variable angegeben. Soll z. B. der Pfad zu den Linux-Kernel-Modulen dargestellt werden, muß beachtet werden, daß dieser Pfad von der Version des Kernel abhängig ist. Für einen Kernel mit der Version 2.4.1 lautet dieser Pfad z. B. `/lib/modules/2.4.1/`
Soll keine konkrete Versionsnummer angegeben werden, kann stattdessen eine kursiv gesetzte Variable eingesetzt werden:
`/lib/modules/kernel-vers/`
Zur Nutzung dieses Pfads muß der kursiv gesetzte Teil `kernel-vers` durch die entsprechende Information (hier: die Versionsnummer des Kernel) ersetzt werden.
- Im Text auftretende Fachbegriffe können im Glossar des Buches nachgelesen werden. Hierzu wird der Begriff im Text wie in folgendem Beispiel dargestellt: →*CTI*.

die Einwahl scheitert. Im Normalbetrieb sollte diese Option nicht aktiviert sein.

Zur Konfiguration der Einwahl des gewünschten Providers kann am einfachsten zunächst eine Kopie von `isdn/avm` mit dem Namen des gewünschten Providers erstellt werden. Anschließend sollten zumindest die Parameter `user`, `password` und `number` angepaßt werden. Anschließend kann ein Einwahlversuch wie oben beschrieben durchgeführt werden.

Als nächstes muß noch das Problem gelöst werden, daß nach dem die Einwahl als Benutzer `root` funktioniert, auch ein normaler Benutzer `pppd` mit den erforderlichen Parametern starten darf. Hierzu kann als eine Lösungsmöglichkeit `/usr/sbin/pppd` SUID-root gemacht werden, wodurch der Prozeß immer als `root` arbeitet, egal wer ihn startet:

```
root@erde:/ # chown root /usr/sbin/pppd
root@erde:/ # ls -l /usr/sbin/pppd
-rwxr-x--- 1 root dialout 206856 Jan 14 17:00 /usr/sbin/pppd
root@erde:/ # chmod u+s /usr/sbin/pppd
root@erde:/ # ls -l /usr/sbin/pppd
-rwsr-x--- 1 root dialout 206856 Jan 14 17:00 /usr/sbin/pppd
```

Durch das `s`-Bit in den Benutzerrechten von `pppd` wird der Prozeß als der Benutzer gestartet, dem die Datei gehört (hier also: `root`). Eine Alternative ist die Verwendung von `sudo`, einem Paket, das es `root` ermöglicht bestimmten Benutzern für bestimmte Kommandos `root`-Rechte zu erteilen. Hierzu muß die Datei `/etc/sudoers` als `root` mit dem Kommando `visudo` wie folgt angepaßt werden. Der Benutzer dem hier das Recht eingeräumt werden soll `pppd` mit `root`-Rechten zu starten sei `tux`:

```
tux        ALL=NOPASSWD: /usr/sbin/pppd
```

Durch diesen Eintrag kann `tux` anschließend z. B. das Kommando `sudo /usr/sbin/pppd call isdn/avm` mit `root`-Rechten ausführen.

2.3 ADSL unter Linux

Neben der Möglichkeit, einen Internet-Zugang mit Hilfe eines Modems oder über eine ISDN-Verbindung herzustellen, bieten immer mehr Provider einen ADSL-basierten Zugang. Da kaum ein Provider, außer der Telekom über ein eigenes Leitungsnetz zum Kunden verfügt, mieten solche Anbieter oftmals Leitungen von der Telekom. In solchen Fällen ist die Konfiguration des Zugangs für diesem Provider oftmals mit der für den von der Telekom als T-DSL vermarkteten Dienst identisch. Unterschiedlich sind dann nur der zu verwendende Benutzername und das Paßwort. Eine gute Übersicht über ADSL-Anbieter für Privatper-

Tabelle 2.4: DSL-Techniken im Überblick

DSL-Technik	Adern-Paare	Upstream	Downstream	Reichweite in km
ADSL ⁷	1	64-640	1500-8000	4.5
T-DSL	1	128/192 ⁸	768/1500 ⁸	4.0
HDSL ⁹	1-3	2048	2048	4.0
SDSL ¹⁰ /SHDSL ¹¹	1/1-3	144-2320	144-2320	3.0
VDSL ¹²	1	2300	53084	0.5

sonen ist unter der URL <http://www.adsl4linux.de/anbieter/> verfügbar. Informationen darüber, in welchen Städten T-DSL verfügbar ist, können auf der Web-Seite der Telekom unter der URL <http://service.t-online.de/t-on/inte/tdsl/ei/CP/ei-tdsl-bestellung.html> bezogen werden.

2.3.1 Grundlagen

Die Familie der DSL⁵-Techniken zu der auch ADSL gehört, ist keine ganz neue Technik. Erste Varianten von DSL wurden in den USA bereits Ende der 80er Jahre erprobt. Allen DSL-Varianten gemein, ist die Tatsache, daß Daten auf ein oder mehreren Kupferadernpaaren übertragen werden. Da fast alle Telefonanschlüsse in Deutschland über Kupferkabel realisiert sind, ist DSL eine Technik für die keine neue Infrastruktur benötigt wird und dennoch eine verglichen mit einem ISDN-B-Kanal mindestens 12fache Geschwindigkeit von 768KBit/sec⁶ und mehr erreicht werden kann. Die letztlich erreichbare Geschwindigkeit hängt dabei sehr stark von der Kabellänge und den elektrischen Eigenschaften der der Kupferadern ab, also letztlich von der Entfernung der Vermittlungsstelle des Providers bis zur Dose des Kunden. Die Grenze für die meisten DSL-Verfahren liegt hier bei 4–5 km. Tabelle 2.4 gibt einen kurzen Überblick über die wichtigsten DSL-Technologien.

Wie in Tabelle 2.4 zu sehen ist, gliedern sich die DSL-Techniken in symmetrische und asymmetrische Verfahren. Symmetrisch meint, daß die Transferrate vom Provider zum Rechner (downstream) genau gleich groß ist wie die Transferrate vom Rechner zum Provider (upstream). Solche Verfahren, wie z. B. HDSL und SDSL/SHDSL sind daher insbesondere für Firmen interessant, bei denen etwa gleich viel Daten gesendet und empfangen werden. Für Privatanwender sind hingegen die asymmetrischen Verfahren (ADSL, T-DSL) von Interesse. Asymmetrisch meint, daß die Transferrate vom Provider zum Rechner (downstream) wesentlich größer ist als die Transferrate vom Rechner zum Provider (upstream). Diese

⁵Digital Subscriber Line

⁶Typischer Wert heute angebotener ADSL-Produkte in Downstream-Richtung, wie z. B. T-DSL

Aufteilung entspricht den Anforderungen der überwiegenden Zahl von Internet-Nutzern, die weitaus mehr Informationen aus dem Internet beziehen (z. B. Surfen im Web) als sie ihrerseits versenden (z. B. Mail).

Ein Vorteil speziell der ADSL/T-DSL-Technik (im Gegensatz zu den synchronen SDSL, HDSL-Techniken) besteht darin, daß neben dem Datenstrom zusätzlich auch Sprachdienste auf den gleichen Leitung übertragen werden können, ohne das beide Dienste sich stören. Dies beruht auf der Tatsache, das Sprach- und digitale Nutzdaten auf unterschiedlichen Frequenzbändern über die Kupferadern übertragen werden. So werden analoge Sprachdaten im Telefonnetz im Frequenzbereich von 100-3200 Hz übertragen. ISDN-Dienste verwenden den Bereich von 4 kHz - 120kHz. Das maximal mögliche Frequenzspektrum für die Datenübertragung auf Kupferadern reicht jedoch bis 1MHz. Dies bedeutet, das Telefonie-Dienste inkl. ISDN nur etwas mehr als 10% der verfügbaren Bandbreite ausnutzen und ca. 90% bisher ungenutzt blieben. Dieser zuvor noch ungenutzte Bereich wird von DSL zur Datenübertragung verwendet. Um die zur Verfügung stehende Bandbreite flexibel verteilen zu können und zur Umgehung von Störsignalen in bestimmten Frequenzbereichen wird der DSL-Frequenzbereich in einzelne, unabhängige Kanäle unterteilt in denen Daten übertragen werden können. Das hierzu verwendete Modulationsverfahren heißt DMT¹³. Um die auf „einem“ Kabel übertragenen verschiedenen Signale beim Kunden wieder voneinander trennen zu können ist ein sogenannter Splitter notwendig, der den niederfrequenten Anteil des Signals (Telefonie, ISDN) vom hochfrequenten Anteil (DSL) trennt. Durch diesen Aufbau ist es dem Besitzer eines kombinierten Telefon/ADSL-Anschlusses problemlos möglich zu telefonieren und gleichzeitig Daten zu übertragen. Dabei spielt es keine Rolle ob der Benutzer über einen ISDN- oder ein analogen Anschluß verfügt.

Für den Einsatz von ADSL wird neben der Leitung zur Vermittlungsstelle des Providers ein DSL-Modem benötigt, das an den Splitter angeschlossen wird. Das DSL-Modem überträgt die von der Vermittlungsstelle empfangenen Daten in Ethernetpakete. Zur Verbindung des Rechners mit dem DSL-Modem dient eine handelsübliche PC-Ethernetkarte. Daten, die vom Rechner gesendet werden, werden über die Netzwerkkarte des Rechners zum Modem gesendet und dort mittels DSL zur Vermittlungsstelle übertragen und weitergeleitet.

Ein Verbindungsaufbau im klassischen Sinn, also das Wählen der Telefonnummer eines Providers ist bei ADSL-Verbindungen nicht erforderlich, da es sich bei ADSL praktisch um eine Standleitung handelt. Da der Provider jedoch die Nutzung der Leitung abrechnen will und die Abrechnung in der Regel nach Nutzungszeit, nicht nach Volumen erfolgen soll, braucht man einen Weg, der es dem Provider gestattet festzustellen, wann und wie lange der Nutzer die ADSL-Leitung verwendet hat. Zu diesem Zweck wurde das PPPoE- (PPP over Ethernet)

¹³Discret Multi-Tone

Protokoll entwickelt. PPPoE ist ein in RFC 2516 (siehe <http://www.faqs.org/rfcs/>) beschriebenes Protokoll und gestattet analog einer normalen Wählverbindung mit PPP, im Wesentlichen die Authentifizierung eines Benutzers (Login und Passwort) sowie die Zuteilung einer IP-Adresse.

Für Linux existieren inzwischen zwei verschiedene Implementierungen des PP-PoE-Protokolls, eine Kernel-basierte Version, sowie der als normaler Benutzerprozeß arbeitende Roaring Penguin-PPPoE-Daemon. Die beiden Lösungsmöglichkeiten werden in den folgenden Abschnitten genau beschrieben.

2.3.2 Einführung in T-DSL

T-DSL ist eine Telekom-Variante der ADSL- (Asymmetric Digital Subscriber Line) Technik. Obwohl immer mehr Provider DSL-Dienste anbieten, ist die Telekom mit ihrem T-DSL-Produkt immer noch Marktführer. Im folgenden wird der Anschluß der notwendigen Hardwarekomponenten, als auch die auf Softwareseite notwendige Konfiguration besprochen. Da viele andere DSL-Anbieter letztlich nur die Leitungen der Telekom mieten und deren Hardware mitbenutzen, gilt die unten stehende Beschreibung auch für einige andere Anbieter. Eine gute Übersicht über die notwendigen Konfigurationsschritte für die verschiedenen DSL-Anbieter ist unter der URL <http://www.adsl4linux.de/> verfügbar.

Da T-DSL eine ADSL-Variante darstellt, können über die bereits vorhandene Anschlußleitung für das Telefon zusätzlich mittels DSL Daten übertragen werden.

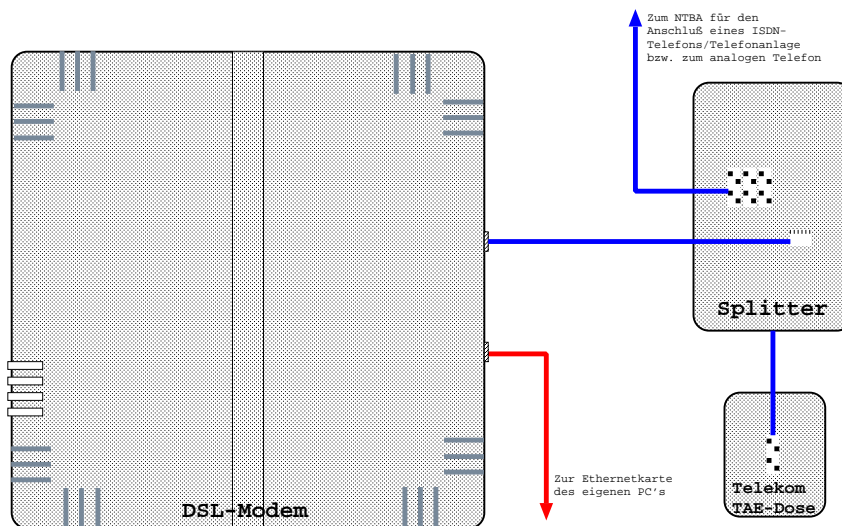


Abbildung 2.16: Anschluß der T-DSL Hardware-Komponenten

Dabei ist es technisch unerheblich, ob der Kunde einen analogen oder über einen ISDN-Anschluß verfügt. Bei der Installation erhält der Kunde von der Telekom einen Splitter und ein als T-DSL-Modem bezeichnetes Gerät. Die Verkabelung dieser Geräte ist schematisch in Abbildung 2.16 auf der vorherigen Seite dargestellt.

An die von der Telekom gesetzte TAE-Anschlußdose, an die bisher das analoge Telefon bzw. der NTBA¹⁴ der Telekom angeschlossen war, wird jetzt der Splitter gesteckt. Der Splitter verfügt über mehrere Ausgänge. Ein Ausgang mit einem RJ45 Stecker wird mit dem Splitter-Eingang des DSL-Modems verbunden. An die TAE-Anschlüsse des Splitters kann bei einem ISDN-Anschluß der NTBA angesteckt werden. Wer noch einen analogen Anschluß hat, kann an die TAE-Stecker ein oder mehrere analoge Geräte (z. B. Telefon und Modem) anschließen. Der Splitter wird also einfach zwischen der Telekom-Anschlußdose und den bisher angeschlossenen Geräten geschaltet.

Die Verbindung des T-DSL-Modems mit einem Rechner erfolgt über eine Ethernet-Schnittstelle. Dies bedeutet, daß der Rechner über eine Ethernet-Karte verfügen und diese auch konfiguriert sein muß, was mit Hilfe der Installationswerkzeuge der jeweiligen Linux-Distribution problemlos durchführbar sein sollte (letztlich muß lediglich das zur Karte passende Kernel-Modul, evtl. mit Parametern, geladen und das Interface mit `ifconfig eth0 up` aktiviert werden). Die Ethernetkarte wird mit dem zweiten RJ45-Anschluß des DSL-Modems verbunden.

Neben der klassischen Möglichkeit zusammen mit dem T-DSL-Anschluß auch das DSL-Modem von der Telekom zu beziehen, existiert seit Anfang 2002 auch die Möglichkeit das Modem von einem Fremdhersteller zu beziehen. Die Hardware des Fremdherstellers sollte dem U-R2 Standard der Telekom entsprechen um Probleme mit der Kompatibilität zwischen Modem und Vermittlungsstelle der Telekom zu vermeiden. Ein Beispiel für eine solche alternativ-Lösung ist die Fritz!Card DSL von AVM. Hierbei handelt es sich um eine Einsteckkarte für den PC, die sowohl als ISDN-Karte als auch als DSL-Modem dient. Zur Installation solcher Produkte sollte der Beschreibung des Herstellers gefolgt werden, um die notwendige Hardware in Betrieb nehmen zu können. Neben solchen „internen“ Lösungen existieren auch weitere externe DSL-Modems. Diese stellen die Verbindung zum Rechner z. B. über eine USB-Schnittstelle her und sind für Linux ungeeignet.

Der Vorteil einer wie in Abbildung 2.16 auf der vorherigen Seite dargestellten Lösung eines externen DSL-Modems mit Ethernet-Schnittstelle, besteht genau darin, das hierzu außer einem Treiber für die Ethernetkarte im Linux-Rechner keine weiteren hardwarespezifischen Treiber benötigt werden, wodurch die Konfiguration vereinfacht wird und man nicht auf die spezielle Treiber des Herstellers (und

¹⁴Bei ISDN-Anschlüssen wird die eigene Telefonanlage bzw. das ISDN-Telefon an den NTBA angeschlossen, der wiederum mit der Telefondose verbunden werden muß.

evtl. Bugs) angewiesen ist. Der für die externe Lösung notwendige Betrieb einer Ethernet-Schnittstelle ist in Linux vollkommen problemlos. Ein weiterer Vorteil der Lösung eines externen DSL-Modems mit Ethernet Schnittstelle besteht im getrennten Betrieb von DSL-Modem und Rechner. Das DSL-Modem muß sich nach dem Einschalten mit der Vermittlungsstelle synchronisieren, was u.U. einige Minuten dauern kann. Steckt das Modem im Rechner muß diese Synchronisierung jedes Mal beim Einschalten des Rechners erfolgen, wodurch es vorkommen kann, das der Rechner schon betriebsbereit ist, das eingebaute Modem aber die Synchronisation mit der Vermittlungsstelle noch nicht abgeschlossen hat. Das externe DSL-Modem kann hingegen unabhängig vom Rechner aktiv bleiben.

Zu Abrechnungszwecken, also um Online-Zeiten der Kunden protokollieren zu können, wird auf der T-DSL-Leitung eine spezielle Variante des PPP-Protokolls, das als PPPoE bezeichnet wird, übertragen. Durch diese Erweiterung von PPP müssen sich Benutzer bei der Einwahl beim Provider anmelden. Hierzu ist auf Client-Seite die klassische PPP-Konfigurationen erforderlich, die eine Anpassung der `/etc/ppp/options`-Datei sowie Einträge zur Authentifizierung in `/etc/ppp/pap-secrets` mit einigen Eigenheiten erfordert.

Das von der Telekom angebotene T-DSL bietet zur Zeit Übertragungsraten von 768 kBit/sec downstream und 128 kBit/sec upstream an. Verglichen mit einem Modem (56.000 Bit/sec) oder einem ISDN-Zugang (max. 128 kBit/sec) sind dies insbesondere downstream geradezu berauschende Übertragungsraten. In einigen Städten ist bereits die neue Variante von T-DSL, T-DSL 1500 verfügbar. In dieser Konfiguration ist ein Download mit einer Geschwindigkeit bis zu 1500kBit/sec (1.5 Mbit/sec) und ein Upload mit bis zu 192 KBit/sec möglich.

Unter Linux existieren verschiedene Möglichkeiten, T-DSL zu nutzen, die unterschiedlich hohen Konfigurationsaufwand erfordern. Die favorisierte Variante basiert auf einem PPPOE-Treiber im Linux-Kernel, die zweite Variante verwendet den Roaring Penguin PPPOE-Treiber, der als normaler Benutzerprozeß arbeitet. Aufgrund der relativ hohen Transferrate, die bei T-DSL möglich ist, sollte mit einem aktuellen 2.4er-Kernel möglichst der Kernel-basierte Treiber eingesetzt werden, da ansonsten die Datenübertragung eine unnötig hohe Systemlast erzeugen kann. In Fällen, wo dies nicht gelingt, oder wenn ein älterer Kernel (Version kleiner als 2.4) eingesetzt wird, kann immer noch RP-PPPOE eingesetzt werden.

2.3.3 T-DSL-Konfiguration mit `pppoe`-Support im Kernel

Seit der Kernel-Version 2.4 gehört das PPPoE-Protokoll fest zum Kernel. Voraussetzung zur Nutzung ist natürlich, daß in der Kernelkonfiguration der PPPoE-Support auch aktiviert wurde. Bei allen neueren SuSE-Distributionen, die mit Kernel 2.4 ausgeliefert werden, ist dies der Fall. Ein Test, ob der PPPoE-Treiber im Kernel aktiviert wurde, ist die Suche nach den Kernel-Modules `pppoe` und

pppoeX im Moduleverzeichnis des Kernels unter `/lib/modules`. Hierzu kann folgendes Kommando verwendet werden:

```
root@erde:/home/tux # find /lib/modules/`uname -r`/ -name 'ppp[oe]*'
/lib/modules/2.4.17/kernel/net/atm/pppoe.o
/lib/modules/2.4.17/kernel/drivers/net/pppoe.o
/lib/modules/2.4.17/kernel/drivers/net/pppox.o
```

Die oben dargestellte Ausgabe bedeutet, daß der Kernel über PPPoE-Support verfügt. Ansonsten muß der Kernel neu konfiguriert, übersetzt und installiert werden bevor DSL genutzt werden kann. Die entsprechenden Einstellungen befinden sich in der Kernelkonfiguration im Abschnitt `Network device support`. Dort muß der Punkt `PPP over Ethernet` als Modul markiert werden. Da der Treiber je nach Kernelversion noch als „Experimental“ gekennzeichnet ist, muß zuvor im Abschnitt `Code maturity level options` der Punkt `Prompt for development and/or incomplete code/drivers` aktiviert werden. Informationen zur Übersetzung eines neuen Kernels finden Sie im Abschnitt 3.2.2.1 auf Seite 213.

Eine weitere Voraussetzung zur Nutzung von T-DSL ist neben einem Kernel, der PPPOE unterstützt auch eine aktuelle Version des `pppd` (mindestens Version 2.4.1) notwendig. Zusätzlich muß der Original `pppd` einen Patch für die PPPoE-Unterstützung enthalten. Bei SuSE Linux ist dies bereits der Fall. Für andere Distributionen sollte die Version und der Patchstand von `pppd` überprüft werden. Die Version des Installierten `pppd` kann am einfachsten herausgefunden werden, indem als Benutzer `root` das Kommando `pppd -version` aufgerufen wird. Um herauszufinden, ob der installierte `pppd` den PPPoE-Patch enthält kann die Datei `pppoe.so` verwendet werden, die in `/usr/lib/pppd/version/` installiert sein sollte, falls `pppd` das Protokoll unterstützt. Falls die Version des installierten Daemons zu alt bzw der PPPoE-Patch nicht integriert ist, muß `pppd` deinstalliert und anschließend von Hand neu übersetzt werden. Der Quellcode von `pppd` ist unter <http://www.sfgoth.com/~mitch/linux/atm/pppoeatm/> verfügbar. Für die Installation genügt normalerweise das Auspacken des Quellcode-Archivs sowie der Aufruf von `./configure` dann `make` und schließlich `make install` als Benutzer `root` aus dem Verzeichnis des Quellcodes heraus.

Da die Verbindung von Rechner zu dem DSL-Modem über Ethernet hergestellt wird, muß der Rechner über eine ungenutzte Ethernetkarte verfügen, die unter Linux konfiguriert sein muß. Die einzig notwendige Konfiguration ist die Verwendung einer freien nicht anderweitig genutzten IP-Adresse (z. B. 192.168.99.1). Hierzu muß das entsprechende Ethernet-Interface (mit `ifconfig`) als `up` konfiguriert wurden. Bei SuSE Linux sollte die Konfiguration des Interface am besten mit Hilfe von `yast` oder `yast2` erfolgen. Wichtig ist, sich zu merken, welches Interface der Karte für DSL zugewiesen wurde. Für die

erste Ethernetkarte im Rechner lautet der Name `eth0`. Steckte bereits eine Ethernetkarte im Rechner, so daß für DSL eine zweite eingebaut werden muß, lautet der Gerätenamen voraussichtlich `eth1` usw.

Die eigentliche Konfiguration kann nun auf zwei verschiedenen Wegen erreicht werden. Unter SuSE Linux ist der beste und einfachste Weg die Verwendung von `yast2`. Diese Vorgehensweise wird im folgenden Abschnitt beschrieben. Wer nicht unter SuSE Linux arbeitet oder einfach mehr über den Konfigurationsprozeß erfahren möchte, kann die DSL-Konfiguration auch manuell durchführen. Dieser Weg wird in Abschnitt 2.3.3.2 auf Seite 80 genau beschrieben.

2.3.3.1 Konfiguration mit `yast2`

Seit SuSE Linux 7.1 kann die DSL-Konfiguration ganz einfach mit Hilfe von `yast2` vorgenommen werden. Um DSL anschließend zu nutzen, muß auf dem Rechner sowohl das `smpppd`-Paket als auch möglichst `kinternet` installiert sein. Mehr zu diesen beiden Werkzeugen steht in Abschnitt 2.4 auf Seite 104. Bei älteren SuSE-Installationen muß zusätzlich das Paket `pppoe` installiert sein, das ab SuSE Linux 7.3 im Paket `ppp` aufgegangen ist.

Konfiguration einer Verbindung mit manueller Einwahl

Für die DSL-Konfiguration muß das Kommando `yast2` als Benutzer `root` ausgeführt werden. Da der komfortabelste Weg mit `yast2` zu arbeiten, der unter



Abbildung 2.17: Netzwerk-Konfiguration in `yast2`

Grundsätzlich empfiehlt es sich, hierzu auf das `faxspool`-System von `mgetty+sendfax` zurückzugreifen, da es hierdurch möglich ist, mehrseitige Dokumente direkt zu versenden. Für die Konfigurationsparameter sollten die in Abbildung 3.14 auf der vorherigen Seite dargestellten Werte verwendet werden.

Mit dieser Konfiguration ist es möglich, von `faxspool` unterstützte Dateien zu versenden, wobei jedoch die Felder für den Firmennamen und den Kommentar nicht beachtet werden.

3.2 Faxen über ISDN

Als das Medium Fax entwickelt wurde, war das Telefonnetz noch analog ausgerichtet, d. h., es konnten nur Sprachlaute oder, allgemeiner gesprochen, Töne innerhalb eines bestimmten Frequenzspektrums übertragen werden, jedoch keine digitalen Daten. Aus dieser Situation heraus wurden Modems entwickelt, die die digitalen Bilddaten, die versendet werden sollten, in analoge Tonsignale umwandelten. Dies wurde mit spezieller Hardware erreicht, die in der Lage war, das sehr zeitkritische Fax-Protokoll zu verarbeiten.

Durch die Entwicklung von ISDN wurde das Telefonnetz in Deutschland digital, wodurch – theoretisch – Modems zur Übertragung digitaler Daten nicht mehr benötigt werden. Eine Seite wird von einem Scanner digitalisiert und kann direkt über das Telefonnetz an eine entsprechende Gegenstelle versendet werden. Nach diesem Prinzip arbeiten Fax-Geräte der Gruppe 4, die jedoch nicht kompatibel zu den verbreitetsten (analogen) Fax-Geräten der Gruppe 3 sind. Ein Gerät der Gruppe 4 kann nur mit einem anderen Gerät der Gruppe 4 kommunizieren. Entsprechendes gilt für Geräte der Gruppe 3.

Um ISDN-Nutzern die Möglichkeit zu geben, ohne ein Modem Faxe (der verbreiteten Gruppe 3) zu versenden oder zu empfangen, wurde daher Software entwickelt, die die Funktionen des Modems emuliert. Die Schwierigkeit bei diesem Verfahren besteht darin, daß diese Aufgabe sehr zeitkritisch ist und einen entsprechend leistungsfähigen Rechner voraussetzt. Die Software muß in der Lage sein, innerhalb weniger Millisekunden auf Nachrichten des anderen Fax-Geräts zu reagieren. Auf einem System wie Linux, auf dem mehrere Benutzer gleichzeitig arbeiten können und u. U. mehrere 100 Prozesse parallel laufen, gestaltet sich eine solche Software-Emulation eines Modems nicht ganz einfach. Der Fax-Prozeß kann nicht selbst bestimmen, wann er rechnen darf und wann er warten muß, um wieder die CPU nutzen zu können. Diese Entscheidung wird vom Linux-Scheduler getroffen, der den verschiedenen Prozessen Rechenzeit in der CPU zuteilt. Dies kann dazu führen, daß der Prozeß mitten in der Bearbeitung einer zeitkritischen Fax-Protokolloperation unterbrochen wird. Auf Systemen, die einem Prozeß gestatten, die CPU solange wie nötig zu belegen, ist die Entwicklung von ISDN-Fax-Software daher wesentlich leichter. Die Kehrseite dieser zu-

nächst positiv scheinenden Eigenschaft ist, daß der nicht zu unterbrechende Prozeß im Fehlerfall leicht den ganzen Rechner zum Absturz bringen kann, was bei einem Linux-System nicht möglich ist.

Einen Ausweg aus diesem Dilemma bietet die Verlagerung der Software, die zur Simulation des Fax-Protokolls dient, auf die ISDN-Karte selbst. Solche, als „aktive“ bezeichneten ISDN-Karten verfügen über eine eigene CPU und entsprechend viel Hauptspeicher, was sie, einmal abgesehen von dem etwa 10mal höheren Preis, von den herkömmlichen passiven ISDN-Karten unterscheidet.

Die Kommunikation zwischen der auf der Karte arbeitenden Software und der auf dem PC arbeitenden Anwendung erfolgt über eine als \rightarrow CAPI (Common-ISDN-Application Programming Interface) bezeichnete Schnittstelle, die letztlich aus einer Bibliothek von Funktionen besteht, die der Anwendung die Steuerung der Karte erlaubt. Darüber hinaus sind Entwicklungen auf dem Weg, die es gestatten werden, eine aktive Karte mit ganz normalen AT-Kommandos (der Fax-Klasse 2) über die `ttYIx`-Gerätedateien anzusteuern. Dies hat den großen Vorzug, daß alle bestehende Modem-Fax-Software, wie z. B. HylaFAX, ohne Anpassung verwendet werden könnte.

Die zur Zeit beste Möglichkeit ist die Nutzung von beliebigen Karten des Herstellers AVM (z. B. Fritz-Card), da AVM für alle seine Karten eine CAPI für Linux bereitstellt, mit deren Hilfe der Faxversand über AVM-Controller möglich wird.

Die zweite Möglichkeit, ein Fax über ISDN zu versenden oder zu empfangen, ist die Nutzung einer der unterstützten aktiven ISDN-Karten. Da sich dieser Bereich immer noch sehr rasch entwickelt, sollte man sich die Linux-Kernel-Dokumentation ansehen, die sich üblicherweise im Verzeichnis `/usr/src/linux/Documentation/isdn` befindet. Von den Dokumenten in diesem Verzeichnis ist insbesondere die Datei `README.fax` interessant, in der die unterstützten Karten im Hinblick auf das Fax-Versenden über den ISDN-Terminalemuulator mit Hilfe der `ttYIx`-Gerätedateien beschrieben werden.

3.2.1 Faxversand über die AVM-CAPI-Lösung

Neben der Möglichkeit, Faxe über ein Modem zu versenden, existiert unter Linux ebenfalls die Möglichkeit, ein Fax mit Hilfe eines ISDN-Controllers zu versenden. Möglich wird dies durch die \rightarrow CAPI von AVM, mit deren Hilfe der Faxversand für alle ISDN-Karten dieses Herstellers realisiert ist. Aufbauend auf der CAPI arbeitet die ebenfalls von AVM entwickelte `capi4hylafax`-Software, die den eigentlichen Faxversand durchführt und eine Anbindung an das unter Linux weit verbreitete Softwarepaket HylaFAX zur Verfügung stellt. Grundsätzlich arbeitet `capi4hylafax` nicht nur mit Karten von AVM zusammen, sondern mit jeder Karte, für der Hersteller eine CAPI (Spezifikation 2.0) bereitstellt. Hier bleibt zu

hoffen, daß neben AVM noch weitere Hersteller eine CAPI für ihre Karten unter Linux zur Verfügung stellen.

In den folgenden Abschnitten wird die Installation und Nutzung des ISDN-basierten Faxdienstes beschrieben. Dabei bieten sich dem Nutzer grundsätzlich zwei Möglichkeiten:

- ❑ Zum einen können die Programme `c2faxsend` und `c2faxrecv` zum Versenden/Empfangen eines Fax verwendet werden.
- ❑ Die zweite Möglichkeit besteht in der Anbindung dieser Applikationen in HylaFAX, so daß der Faxversand bzw. der Empfang eines Fax, wie bereits in Abschnitt 3.1.4 auf Seite 119 beschrieben, ganz normal mit den Mitteln von HylaFAX, jetzt aber über einen ISDN-Controller, durchgeführt werden kann. In diesem Fall können aber nach der Installation Faxe nur noch über die ISDN-Karte versendet und empfangen werden und nicht mehr mit einem zuvor von HylaFAX genutzten Modem.

3.2.1.1 Installation und Konfiguration

Zur Installation muß zunächst die AVM-CAPI-Bibliothek selbst installiert werden. Dieser Vorgang ist ausführlich in Abschnitt 2.2.4 auf Seite 48 beschrieben. Wenn diese Installation erfolgreich beendet wurde, so daß der AVM-Testserver angewählt werden konnte, kann mit der eigentlichen Installation von `capi4hylafax` begonnen werden.

Ab SuSE Linux Version 8.1 ist die Software bereits in dem fertig übersetzten RPM-Paket `capi4hylafax` enthalten. In diesem Fall muß lediglich das Paket selbst z. B. mit `yast2` installiert werden. Wer `capi4hylafax` jedoch mit HylaFAX integrieren will, sollte u. U. dennoch die Installation vom Quelltext wählen, da hier alle notwendigen Schritte der Integration der CAPI in HylaFAX automatisch durchgeführt werden. Ansonsten muß diese Integration manuell durchgeführt werden. Hinweise hierzu stehen auf Seite 201. Ohne diese Integration können die CAPI-Programme lediglich für sich, also ohne HylaFAX verwendet werden. Für alle andere Distributionen muß der Quellcode selbst übersetzt werden, was jedoch nicht sonderlich schwer ist. In diesem Fall besteht der erste Schritt im Download der Software vom FTP-Server von AVM `ftp://ftp.avm.de`

Das `capi4hylafax`-Paket von AVM kann direkt vom FTP-Server, der unter URL `ftp://ftp.avm.de/tools/` zu finden ist, heruntergeladen werden. In dem oben genannten Verzeichnis befinden sich weitere Unterverzeichnisse, die für den jeweiligen ISDN-Kartentyp stehen, wie z. B. `a1` oder `fritzcard` für die ISA-Fritz-Card und `b1` für den aktiven AVM-B1-Controller. In dem kartenspezifischen Unterverzeichnis befindet sich dann ein Verzeichnis `linux`, in

dem die gewünschte Software liegt. Der Name des capi4hylafax-Pakets ist capi4hylafax-01.01.02.tar.gz.

Nach dem Download muß das Archiv ausgepackt werden. Anschließend wechselt man in das aus dem Archiv stammende neue Unterverzeichnis, das wiederum in Abhängigkeit vom Kartentyp benannt ist:

```
root@erde:/tmp # tar xvf /tmp/capi4hylafax-01.01.02.tar.gz
capi4hylafax-01.01.02/
capi4hylafax-01.01.02/bin_SuSE6.3_6.4_7.0/
capi4hylafax-01.01.02/bin_SuSE6.3_6.4_7.0/c2faxsend
capi4hylafax-01.01.02/bin_SuSE6.3_6.4_7.0/c2faxrecv
capi4hylafax-01.01.02/src/
...

root@erde:/tmp # cd capi4hylafax-01.01.02
root@erde:/tmp/capi4hylafax-01.01.02 # ls -CF
AUTHORS                Makefile.in           fritz_pic.tif
COPYING                Readme_CAPI4HylaFAX  install*
CVS/                   Readme_src            sample_AVMC4_config.faxCAPI*
ChangeLog              aclocal.m4            sample_faxrcvd*
GenerateFileMail.pl*  config.faxCAPI       src/
Liesmich_CAPI4HylaFAX configure*
Makefile.am           configure.in
```

Die Pakete von AVM sind auf die SuSE Linux-Distribution zugeschnitten. Auf SuSE-Systemen hat man daher keine Probleme, sie zu installieren. Jedes Paket enthält bereits alle Programme fertig übersetzt. Der Installationsprozeß ist durch eine Datei, die Teil des jeweiligen Pakets ist, ausführlich dokumentiert. Die Datei heißt Readme_CAPI4HylaFAX bzw. Liesmich_CAPI4HylaFAX. Ein Blick in diese Dokumentation lohnt sich auf jeden Fall.

Im Normalfall besteht die Installation lediglich aus dem Aufruf des Skripts install. Lediglich wenn diese Installation fehlschlägt (z. B., weil Bibliotheken fehlen) oder falls die Software auf nicht SuSE Linux-System eingesetzt wird, sollte das eigene Übersetzen der Software in Erwägung gezogen werden. Auch für diesen Fall enthält das Archiv eine Dokumentation. Die Datei heißt Readme_src.

Im folgenden wird zunächst die normale Installation ohne eigenes Übersetzen beschrieben. Anschließend wird auf die Installation mit übersetzen des Quellcodes eingegangen.

Normale Installation von capi4hylafax

Die Installation muß als Benutzer root gestartet werden. Je nach Verwendung kann die Installation so ausgeführt werden, daß capi4hylafax anschließend

te nicht wie `pop` in Klartext überträgt. Leider unterstützen die meisten Provider lediglich das POP-Protokoll. Der nächste wichtigste Konfigurationsschritt ist die Angabe des Mail-Servers des Providers mit der Konfigurationsvariable `server`. Dazu gehört natürlich auch der Benutzername samt Passwort für den Mailzugang des Providers, die mit den Konfigurationsvariablen `user` und `pass` bestimmt werden. Damit `masqmail` weiß, wohin die vom Provider geholte Mail versendet werden soll, muß eine gültige Zieladresse mit der Variablen `address` angegeben werden. Im Normalfall wird dies eine lokale Mail-Adresse sein, etwa `okurth@localhost`. Prinzipiell ist es jedoch auch möglich, hier eine externe Mailadresse anzugeben, an die die Mail gesendet wird, auch wenn eine solche Konfiguration in der Regel kaum Sinn macht.

Mit Hilfe der Konfigurationsoption `do_keep`, die entweder `true` oder `false` sein kann, wird festgelegt, ob abgerufene Mail auf dem Server des Providers belassen (`true`) oder gelöscht wird (`false`). Die Mail dort zu belassen, macht nur dann Sinn, wenn man noch von einem anderen Rechner aus darauf zugreifen möchte. In diesem Fall sollte man aber darauf achten, daß auf irgendeinem System die geholte Mail vom Server des Providers gelöscht wird, da die Mailbox beim Provider ansonsten immer weiter bis zu einer evtl. durch den Provider gesetzten Maximalgrenze wächst, was schließlich dazu führt, daß, wenn die Grenze überschritten wird, keine neue Mail mehr empfangen werden kann. Zusätzlich sollte in diesem Fall die Variable `do_uidl` auf `true` gesetzt werden, was verhindert, das einmal von einem System abgerufene Mail beim nächsten Verbindungsaufbau wieder neu geholt wird. Um lange Download-Zeiten für sehr große Mails zu verhindern, kann schließlich festgelegt werden, daß Nachrichten nur bis zu einer mit `max_size` festgelegten Maximalgröße geholt werden. Ein Wert von 0 hat die Bedeutung von „kein Limit“.

Der eigentliche Abruf der Mail muß vom Benutzer angestoßen werden, indem auf der Kommandozeile von `masqmail` die Option `-g` gefolgt vom Namen des Providers angegeben wird. Ist der Name der `get`-Konfigurations-Datei z. B. `my-provider.get`, so muß der Aufruf entsprechend `masqmail -g myprovider` lauten. Ohne den Namen des Providers werden alle `get`-Konfigurationen durchlaufen, die in der Grundkonfiguration angegeben sind. Die einfachste Lösung ist die Angabe der `-g`-Option zusammen mit dem Aufruf von `masqmail` zum Versenden von Mail in der Datei `/etc/ppp/ip-up`, wie in dem Beispiel auf Seite auf Seite 389. Wird `masqmail` zum Abrufen von Mail verwendet, sollte natürlich der in dem Beispiel stehende Aufruf von `fetchmail` auskommentiert werden.

4.7 Aufbau eines PPP-Servers

PPP (Point to Point Protocol) ist ein Protokoll, das dazu verwendet wird, zwei über eine Leitung verbundene Rechner zu vernetzen. Die „Leitung“ kann eine

mit zwei Modems aufgebaute Telefonverbindung oder eine direkte Verbindung zweier Rechner sein (sei diese mit einem Nullmodem-Kabel oder über IrDA-konforme Geräte, d. h. per Infrarot realisiert; siehe dazu Abschnitt 7.2 auf Seite 628). Auch digitale, z. B. ISDN-basierte Rechner-Verbindungen gehören hierher.

PPP ist ein sehr flexibles Protokoll, das es erlaubt, eine Vielzahl von Parametern, z. B. die IP-Adressen der beiden Kommunikationspartner, dynamisch auszuhandeln. Darüber hinaus unterstützt PPP verschiedene Verfahren (\rightarrow PAP und \rightarrow CHAP) zur Authentifizierung der Kommunikationspartner. Aufgrund dieser Flexibilität hat PPP sehr große Verbreitung gefunden, und fast jeder ISP bietet seinen Kunden einen Zugang über PPP an. Andere Protokolle wie etwa \rightarrow SLIP (Serial Line IP) wurden meist zugunsten von PPP aufgegeben.

4.7.1 Voraussetzungen

Zur Verwendung von PPP muß die entsprechende Software installiert sein. Hierzu gehören insbesondere der `pppd` sowie weitere Hilfprogramme, wie z. B. das `chat`-Programm, mit dessen Hilfe die Anwahl zum Provider über ein Modem abgewickelt werden kann. Auf SuSE Linux-Systemen muß hierzu das Paket `ppp` installiert werden. Eine weitere Voraussetzung ist die Aktivierung des PPP-Kernel-Support. Hier muß in der Datei `/usr/src/linux/.config` eine Zeile `CONFIG_PPP=m` existieren, der kein #-Zeichen vorangestellt sein darf. Gibt es eine solche Zeile bereits, kann davon ausgegangen werden, daß der Kernel PPP unterstützt, andernfalls müssen die Konfigurationsdatei angepaßt und der Kernel neu übersetzt und installiert werden. Eine entsprechende Kurzanleitung wurde bereits in Abschnitt 3.2.2.1 auf Seite 213 gegeben.

PPP verwendet eine ganze Reihe weiterer Protokolle. Die Basis stellt das ISO Protokoll \rightarrow HDLC, das High-Level Data Link Control Protocol, dar⁶. Mit Hilfe dieses Protokolls erfolgen Datenübertragung und Fehlerkontrolle. Dabei können nicht nur IP-Pakete, sondern auch beispielsweise IPX- oder Appletalk-Pakete übertragen werden. Dieser Teil des Protokolls ist in Linux in Form eines Kernel-Moduls implementiert. Die weiteren PPP-spezifischen Protokolle sind in Form des `pppd` implementiert.

Auf HDLC aufbauend, arbeitet \rightarrow LCP, das Link Control Protocol. Es dient im Rahmen von PPP der Einrichtung, Konfiguration und Überprüfung einer Verbindung. Über dieses Protokoll wird beispielsweise die Authentifizierungsmethode ausgehandelt oder auch der MRU- (Maximum Receive Unit) Wert, der die maximale Größe eines zu übertragenden Datagramms beschreibt.

⁶HDLC wird normalerweise nur für synchrones PPP verwendet, wie es z. B. im ISDN PPP-Daemon `pppd` implementiert ist. Um HDLC auch auf asynchronen seriellen Leitungen einsetzen zu können, wird eine standardisierte Modifikation von HDLC verwendet (standardisiert in 3309:1984/PDAD1 als Modifikation zu ISO 3309-1979).

Zur Übertragung der verschiedenen Netzwerkprotokolle wie z. B. IP oder IPX über eine PPP-Verbindung wird eine weitere Familie von Protokollen verwendet, die als $\rightarrow NCP/main$ (Network Control Protocol) bezeichnet wird. Für die Übertragung von IP-Paketen über PPP wird hierzu das $\rightarrow IPCP$ (Internet Protocol Control Protocol) eingesetzt, über das z. B. die Aushandlung der IP-Adressen zwischen den beiden Systemen erfolgt. Darüber hinaus dient IPCP dazu, Einigung in Bezug auf Kompressionsverfahren zwischen den beiden Maschinen zu erzielen. PPP unterstützt zu diesem Zweck die Van-Jacobson Header Compression (kurz VJ-Compression), mit deren Hilfe die Header von TCP-Paketen komprimiert und somit die zu übertragenden Datenmengen reduziert werden können.

Der Aufbau einer PPP-Verbindung besteht zunächst darin, daß eine serielle Verbindung – z. B. über das Telefonnetz mit Hilfe von Modems oder über ein Nullmodem-Kabel zwischen zwei Rechnern – aufgebaut wird. Anschließend wird `pppd` gestartet, der die weitere Kommunikation über diese Leitung steuert. Alle Daten, die über diese Leitung laufen, werden von `pppd` entgegengenommen, überprüft und weitergeleitet. Direkt nach dem Start von `pppd` werden mit Hilfe der genannten Protokolle die Authentifizierung der beiden Rechner sowie die Aushandlung der zu verwendenden IP-Adressen und weiterer Parameter ausgeführt. Jeder der beiden Rechner kann dazu über verschiedene Optionen konfiguriert werden, beliebige Adressen, oder aber nur eine ganz bestimmte statische Adresse zu akzeptieren. Können sich beide Rechner nicht einigen, wird die Verbindung schließlich abgebrochen. Gleiches gilt auch für die Authentifizierung. Jeder Rechner kann konfiguriert werden, nur ein bestimmtes Protokoll zur Authentifizierung zuzulassen, oder aber ein von der Gegenseite vorgeschlagenes Protokoll zu akzeptieren. Kann hier kein gemeinsames Protokoll gefunden werden, das beide Rechner akzeptieren, wird die Verbindung ebenfalls abgebrochen. Wie bereits ersichtlich wird, gibt es bei PPP keine Einteilung in Server- und Client-System. Beide an der Verbindung beteiligten Rechner sind gleichwertig. Zur besseren Übersicht wird im folgenden dennoch der Begriff Client für die Maschine verwendet, die die Verbindung initiiert. Das System, das die Verbindung entgegennimmt, soll als Server bezeichnet werden.

Falls sich beide Rechner über die zu verhandelnden Parameter einig werden konnten, legt `pppd` auf jedem ein neues Netzwerk-Interface an. Das erste Interface hat den Namen `ppp0`, das zweite `ppp1` usw. Diesem Interface wird die zuvor ausgehandelte IP-Adresse zugewiesen, so daß jetzt IP-Daten darüber versendet und empfangen werden können, die über `pppd` zum jeweils anderen Rechner weitergeleitet werden.

Die Einsatzgebiete von PPP sind vielfältig. So können mit Hilfe dieses Protokolls zwei Rechner und damit evtl. auch an diese Rechner gekoppelte Netzwerke miteinander verbunden werden. Darüber hinaus bietet PPP auch einen einfachen Weg, um z. B. auf Daten eines Windows-Rechners zugreifen zu können. Hierzu muß einfach ein PPP-Server auf der Linux-Seite konfiguriert werden. Mit Hil-

fe des Windows DFÜ Netzwerks kann der Windows-Client anschließend eine Verbindung zum Linux-Server herstellen. Die physikalische Verbindung erfolgt am einfachsten über ein Nullmodem-Kabel oder, falls möglich, über eine IrDA-Verbindung (siehe auch Abschnitt 7.2 auf Seite 628).

Die sicherlich häufigste Verwendung von PPP ist die Einwahl bei einem ISP, um Zugang zum Internet zu erhalten. In der Regel erhält der Client vom Server des ISP eine dynamische IP-Adresse zugewiesen. Dynamisch bedeutet, daß diese Adresse bei jeder Einwahl zum ISP wechseln kann. Ein ISP verfügt üblicherweise über hunderte Einwahlpunkte. Jedem dieser Einwahlpunkte (z. B. ein Modem) ist eine bestimmte IP-Adresse über PPP zugeordnet. Wählt sich ein Benutzer bei einem ISP ein, wird er automatisch zum nächsten verfügbaren Anschluß weitergeleitet und erhält die diesem Gerät zugeordnete IP-Nummer. Die Alternative zur dynamische Adreßvergabe ist die statische Adreßvergabe, die jedoch so gut wie gar nicht angewendet wird. In diesem Fall erhält ein Client stets die gleiche IP-Nummer zugewiesen.

Die Vergabe dynamischer Adressen durch den PPP-Server des ISP stellt in der Regel kein Problem dar. Problematisch ist die Verwendung von Diensten, die darauf bauen, daß der Client-Rechner stets unter einer festen IP-Adresse erreichbar ist. Daher ist die direkte Zustellung von Mail zu einer über PPP mit dynamischer Adressvergabe angebotenen Maschine kaum möglich. Auch der Zugriff auf einen Web- oder FTP-Server des Client-Rechners über das Internet ist schwierig, da dieser ja nicht permanent an das Internet angebunden ist und bei jeder Verbindung eine andere IP-Adresse hat. Für das Surfen im WWW und das Abrufen von Mail hingegen stellt dies jedoch kein Problem dar.

4.7.2 Konfigurationsdateien

PPP ist ein im hohen Maße konfigurierbares Protokoll. Zahlreiche Optionen stehen dem Benutzer zur Verfügung, um das Verhalten von `pppd` zu steuern. Grundsätzlich gibt es keine Unterscheidung zwischen der Konfiguration eines PPP-Servers und der Konfiguration eines Client. Für beide werden dieselben Optionen eingesetzt.

Wichtig bei der Arbeit mit PPP ist das Verständnis, wie und in welcher Reihenfolge Optionen verarbeitet werden. Zum einen bietet `pppd` die Möglichkeit, Optionen auf der Kommandozeile zu übergeben. Noch bevor diese Optionen verarbeitet werden, versucht `pppd`, weitere Optionen aus verschiedenen Konfigurationsdateien zu lesen. Hierzu zählen zum einen die globale PPP-Optionsdatei `/etc/ppp/options`, die Benutzerdatei `~/.ppprc` wie auch eine gerätespezifische Datei `/etc/ppp/options.device` (z. B. `/etc/opt/options.ttyS0`).

Die Reihenfolge, in der alle angegebenen Optionen verarbeitet werden, ist wie folgt definiert:

1. `/etc/ppp/options`
Angabe grundlegender Optionen.
2. `$HOME/.ppprc`
Angabe benutzerspezifischer Optionen (nicht für alle Optionen möglich).
3. Optionen auf der Kommandozeile und über Dateien mit der `call`-Option.
Ermöglicht die Angabe spezieller Optionen für einen bestimmten Aufruf. Mit Hilfe der `call`-Option können Optionen z. B. für einen bestimmten Provider wiederum in einer Datei unter `/etc/ppp/peers/` zusammengefaßt werden, die der `call`-Option als Argument übergeben wird.
4. `/etc/ppp/options.device`
Angabe von gerätespezifischen Optionen; wird auf einer PPP-Server-Maschine oftmals für die Angabe der IP-Adressen für den diesem Gerät zugeordneten Client/Server verwendet.
5. `/etc/ppp/ppp-servername`
Angabe von Optionen, die für bestimmte PPP-Server gelten sollen; `ppp-servername` ist dabei der Name dieses Servers. Auf diese Weise hat der Systemadministrator die Möglichkeit, Benutzern für bestimmte Server weitere Rechte einzuräumen.

Werden gleiche Optionen mit verschiedenen Werten angegeben, überschreibt eine in dieser Reihenfolge weiter unten angegebene Option den bisherigen Wert. Aus dieser Reihenfolge ergibt sich auch die grundsätzliche Verwendung der Optionsdateien.

Die Basiskonfiguration eines PPP-Client/Servers sollte in der Datei `/etc/ppp/options` eingetragen werden. Auch der Benutzer hat grundsätzlich die Möglichkeit, PPP-Optionen in seiner `.ppprc`-Datei einzutragen. Diese Möglichkeit ist jedoch eingeschränkt, da ein Benutzer andernfalls durch Umsetzen von Optionen Schaden anrichten könnte. Daher unterscheidet `pppd` zwischen privilegierten und normalen Optionen. Diese Beziehung ist jedoch nicht statisch, sondern abhängig davon, ob in einer der nicht-benutzerspezifischen Dateien die Option `auth` bzw. `noauth` verwendet wurde. Diese Option bestimmt, ob ein Rechner beim Verbindungsaufbau von seinem Gegenüber eine Authentifikation verlangt, bevor Pakete zwischen den beiden Rechnern übertragen werden können. Wurde z. B. vom Systemadministrator die Option `noauth` gesetzt, wird vom System, das der Benutzer anwählen kann, keine Authentifizierung gefordert. Durch die Angabe von `noauth` werden jedoch andere Optionen privilegiert. Hierzu zählt z. B. die Option zur Angabe des Gerätenamens (z. B. `/dev/ttyS1`), über das die Verbindung aufgebaut wird. Der Effekt der Angabe von `noauth` ist, daß der Benutzer jetzt keinen Einfluß mehr auf die Auswahl des Geräts hat, über das die Verbindung aufgebaut wird. Da die Option zur Auswahl der Gerätedatei jetzt privilegiert ist, ist es dem Benutzer nicht mehr gestattet, das Gerät durch die

Angabe einer entsprechenden Option in dem vom Administrator vorgegebenen Wert zu ändern.

Einen Sonderstatus hat die Option `call name`, wenn sie auf der Kommandozeile von PPP angegeben wird. Hierdurch werden weitere Optionen aus der Datei `name`, die im Verzeichnis `/etc/ppp/peers/` gesucht wird, eingelesen. Diese Datei darf privilegierte Optionen enthalten. In der Regel werden in durch `call` bezeichneten Dateien Optionen zusammengefaßt, die für einen bestimmten Provider benötigt werden.

4.7.3 Ein einfacher PPP-Client

Bevor genauer auf eine Server-Konfiguration eingegangen wird, soll an dieser Stelle zunächst eine einfache PPP-Client-Konfiguration beschrieben werden. Die Realisierung dieser Konfiguration und der zugehörigen Start-Skripten erfolgt dabei mit grundlegenden Mitteln. Diese Konfiguration kann auch mit Hilfe komfortablerer Werkzeuge wie `kppp` oder `wvdial` durchgeführt werden, auf die bereits in Abschnitt 2.1.5.4 auf Seite 29 bzw. Abschnitt 2.1.5.2 auf Seite 20 eingegangen wurde. Unter SuSE Linux wird normalerweise eine entsprechende Verbindung einfach mit `yast2` konfiguriert, wodurch die notwendigen Konfigurationsoptionen für den `pppd`-Aufruf erzeugt werden. Im Normalfall kommt der Benutzer also gar nicht direkt mit der PPP-Konfiguration in Berührung. Zum Verständnis, wie PPP funktioniert, ist die folgende Darstellung jedoch sehr hilfreich.

In diesem Beispiel soll davon ausgegangen werden, daß PPP für die Einwahl über ein Modem an einem ISP konfiguriert werden soll, der IP-Adressen dynamisch vergibt und ausschließlich das PAP-Protokoll (siehe Abschnitt 4.7.4.1 auf Seite 369) zur Authentifizierung verwendet. Die Einwahlnummer des ISP sei 02345 8442. Weiterhin habe der ISP eine Kennung `pclient` und ein Passwort `mysecret` zur Verfügung gestellt.

Basis für eine PPP-Verbindung ist zunächst eine entsprechende Konfigurationsdatei. Für dieses Beispiel könnte die Konfiguration in `/etc/ppp/options` abgelegt sein und wie folgt aussehen:

```
# Benutzername fuer die Authentifikation
user pclient
# Nur PAP verwenden, nicht CHAP
-chap
# ISP--System muß sich nicht beim lokalen System authentifizieren
noauth
# Das lokale System akzeptiert IP-Vorgaben vom ISP
ipcp-accept-local
ipcp-accept-remote
# Erzeuge Debug-Output (normalerweise in /var/log/messages)
```



```

sent [CCP ConfAck id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
rcvd [IPCP ConfAck id=0x1 <addr 192.168.50.1> <compress VJ 0f 01>]
local  IP address 192.168.50.1
remote IP address 192.168.50.2
Script /etc/ppp/ip-up started (pid 13276)
rcvd [CCP ConfAck id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
Deflate (15) compression enabled
Script /etc/ppp/ip-up finished (pid 13276), status = 0x0

```

Anhand dieser Ausgaben kann die Aushandlung der Verbindungsparameter und insbesondere der IP-Adressen ersehen werden. Am Ende ist erkennbar, daß das `ip-up`-Skript ausgeführt wurde. Ein Verbindungstest kann nun am einfachsten mit Hilfe des `ping`-Kommandos erfolgen, indem auf Rechner A das Kommando `ping 192.168.50.2` eingegeben wird. Die Ausgabe sollte wie folgt aussehen:

```

root@erde:/root # ping 192.168.50.2
PING 192.168.50.2 (192.168.50.2) from 192.168.50.1: 56(84) bytes of data.
64 bytes from 192.168.50.2: icmp_seq=1 ttl=255 time=2.81 ms
64 bytes from 192.168.50.2: icmp_seq=2 ttl=255 time=2.80 ms
64 bytes from 192.168.50.2: icmp_seq=3 ttl=255 time=2.71 ms
...

```

Anschließend kann z. B. von Rechner A eine FTP-Sitzung zu Rechner B durch Aufruf von `ftp 192.168.50.2` gestartet werden.

4.8 Einrichten eines FTP-Servers

FTP ist ein Protokoll, das der Dateiübertragung zwischen zwei Rechnern in der Regel über das Internet dient. Eine FTP-Verbindung wird von einem Client-Programm, wie etwa `ncftp` (siehe Abschnitt 4.1.2 auf Seite 247), zu einem FTP-Server aufgebaut. Der Server stellt dem Client die Möglichkeit zur Verfügung, Dateien sowohl vom Server zum Client (Download) als auch umgekehrt (Upload) zu transferieren. Verfügt ein Rechner über keine permanente Anbindung zum Internet, kann zur Bereitstellung von Dateien neben dem FTP-Server ein PPP-Server (siehe Abschnitt 4.7 auf Seite 359) eingerichtet werden, der es anderen Benutzern (Rechnern) ermöglicht, über ein Modem etc. auf den FTP-Server zuzugreifen.

Beim Aufbau einer Verbindung von einem Client zum FTP-Server fordert der Server vom Client eine Authentifizierung. Hierbei müssen vom Client – wie beim normalen Login – ein Benutzername und ein zugehöriges Passwort angegeben werden. Eine besondere Stellung hat hierbei der Benutzername `anonymous` (oft auch `ftp`), mit dessen Hilfe ein Download (je nach Serverkonfiguration auch ein Upload) anonym, also ohne Authentifikation, durchgeführt werden kann.

Für den Benutzer `anonymous` wird als Passwort entsprechend einer allgemeinen Konvention die Mail-Adresse des Client angegeben. Dies soll dem Server-Administrator einen Überblick darüber geben, wer auf den von ihm administrierten Server zugreift. Bei manchen Servern muß eine Mail-Adresse angegeben werden, andere Server akzeptieren auch andere „beliebige“ Eingaben für das `anonymous`-Passwort. Da durch den `anonymous`-Zugang wirklich jedem Benutzer ein Zugang zu dem FTP-Server gewährt wird, sind die Möglichkeiten, die diese Kennung hat, sehr beschränkt. In der Regel werden lediglich das Ansehen eines Verzeichnisses `/pub` sowie der Download von Dateien aus diesem Verzeichnis möglich sein. Falls der Server-Administrator einen Upload von Dateien erlaubt, kann dies üblicherweise nur in das Verzeichnis `/incoming` erfolgen. Viele FTP-Server bieten ausschließlich den `anonymous`-Zugang und keinen Benutzerspezifischen.

Der FTP-Server besteht letztlich aus einem Programm `ftpd` (je nach verwendetem Server auch mit anderem Präfix, wie z. B. `wu.ftpd` für den von der Washington University entwickelten FTPD) und einigen Hilfsprogrammen, die es dem FTP-Client z. B. erlauben, Verzeichnisse aufzulisten oder Dateien zu komprimieren. Manche FTP-Server-Distributionen verzichten bewußt auf solche Zusatzprogramme, um dadurch einige Sicherheitsrisiken prinzipiell auszuschließen (wie z. B. der `proftpd`).

Im Laufe der Zeit haben sich verschiedene Varianten von FTP-Servern entwickelt, die mehr Sicherheit und einen erweiterten Funktionsumfang bieten. Die wichtigsten heute existierenden und frei verfügbaren FTP-Server für Linux sind sicherlich WUFTP, PROFTP und der speziell auf Sicherheit ausgelegte VSFTPD, wobei der erstgenannte die große Zahl an Leistungsmerkmalen aufweist und insgesamt eine sehr große Verbreitung gefunden hat⁹. Im folgenden soll die Konfiguration des FTP-Servers `wu.ftpd` beschrieben werden. Obwohl die grundlegende Konfiguration aller FTP-Server sehr ähnlich ist und im wesentlichen die gleichen Schritte erfordert, bestehen doch Unterschiede bei den Leistungsmerkmalen der einzelnen Server und bei den verwendeten Konfigurationsdateien. Wer jedoch die im folgenden, am Beispiel des `wu.ftpd` dargestellten Schritte prinzipiell verstanden hat, sollte keine größeren Schwierigkeiten haben, einen beliebigen anderen Server zu konfigurieren.

4.8.0.1 Funktionsweise

Der WUFTP-Server besteht im wesentlichen aus dem Programm `wu.ftpd`, dem FTP-Daemon. Für jeden Benutzer, der sich bei dem FTP-Server anmeldet, wird ein neuer `wu.ftpd`-Prozeß gestartet, der die Anfragen dieses Benutzers bedient. Der Start des Prozesses erfolgt in der Regel mit Hilfe des `inetd`, einem speziel-

⁹Wer jedoch weniger auf die Zahl der Features als auf möglichst große Sicherheit wert legt, sollte sich VSFTPD ansehen.

len Prozeß, der auf verschiedenen (in `/etc/inetd.conf` konfigurierten) Ports auf Versuche anderer Prozesse (z. B. eines FTP-Client), eine Netzwerkverbindung aufzubauen, wartet. Zur Bearbeitung einer Anfrage wird ein dem jeweiligen Protokoll zugeordnetes Programm gestartet. Im Fall einer FTP-Anfrage, die in der Regel auf Port 21 erfolgt, wird automatisch ein FTP-Server-Prozeß gestartet. Ein Auszug aus der Datei `/etc/inetd.conf` sieht wie folgt aus:

```
# See "man 8 inetd" for more information.
#
# If you make changes to this file, either reboot your machine or
# send the inetd a HUP signal:
# Do a "ps x" as root and look up the pid of inetd. Then do a
# "kill -HUP <pid of inetd>".
# The inetd will re-read this file whenever it gets that signal.
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
ftp      stream  tcp      nowait  root    /usr/sbin/tcpd  wu.ftpd -a -l
# ...
Telnet   stream  tcp      nowait  root    /usr/sbin/tcpd  in.Telnetd
login    stream  tcp      nowait  root    /usr/sbin/tcpd  in.rlogind
talk     dgram   udp      wait    root    /usr/sbin/tcpd  in.talkd
# ...
```

Die hier interessante Zeile ist der `ftp`-Eintrag. Durch diese Zeile wird festgelegt, daß bei einer eingehenden `ftp`-Verbindung der Prozeß `wu.ftpd` als Benutzer `root` mit den angegebenen Argumenten gestartet wird, um die eingehende Verbindung zu bearbeiten. Die Option `-a` weist `wu.ftpd` an, die Konfigurationsdatei `/etc/ftpaccess` zu verwenden, durch die das Verhalten des Servers konfiguriert werden kann (s. u.). Die Option `-l` bewirkt, daß FTP-Verbindungen mit Hilfe des Syslog-Mechanismus protokolliert werden (z. B. in der Datei `/var/log/messages`).

Der links stehende Service-Name `ftp` (wie auch alle anderen Service-Namen) sind wiederum in der Datei `/etc/services` festgelegt. Dort steht für jeden Service die Port-Nummer, auf der Verbindungen für dieses Protokoll erwartet werden, und welches Protokoll für diesen Service verwendet wird. Der Eintrag für den Service `ftp` lautet z. B.:

```
# ...
ftp      21/tcp
# ...
```

Durch diese Angaben weiß `inetd` genau, auf welchen Ports des Rechners, auf dem er arbeitet, auf eingehende Verbindungen gewartet werden soll. Durch `inetd.conf` ist festgelegt, welche Programme die jeweiligen Protokolle verarbeiten können.

Auch `wu.ftpd` wird über den `inetd` gestartet. Daher sollte überprüft werden, ob im Anschluß an die Installation der Software (siehe unten) der Eintrag in `/etc/inetd.conf` vorhanden und nicht durch ein davor stehendes `#`-Zeichen unwirksam gemacht ist. Muß die Datei verändert werden, sollte anschließend ein HUP-Signal an den Prozeß `inetd` gesendet werden, um ihn zu veranlassen, seine Konfiguration neu einzulesen. Dies kann als Benutzer `root` wie folgt durchgeführt werden:

```
root@erde:/root # ps axuw | grep inetd
root    222  0.0  0.0  912    84  ?   S    10:09   0:00 /usr/sbin/inetd
root@erde:/root # kill -HUP 222
```

Nach dem Start des FTP-Daemon erwartet der Server die Anmeldung des Client. Falls der Benutzer, der sich über FTP anmelden möchte, eine lokale Kennung auf dem FTP-Server hat und der Server entsprechend konfiguriert wurde, kann sich der Benutzer mit seinem Kennungsnamen und seinem Passwort anmelden. Anschließend sieht er in der Regel den Inhalt seines Home-Verzeichnisses und kann Dateien von und in dieses Verzeichnis übertragen. Oftmals wird die Konfiguration des Servers jedoch so gestaltet sein, daß sich nur der Benutzer `anonymous` bzw. `ftp` anmelden kann, wobei als Passwort die Client-Mail-Adresse verwendet wird. In diesem Fall werden vom FTP-Server besondere Maßnahmen ergriffen, durch die die Sicherheit des Server-Systems garantiert werden soll. Der FTP-Daemon führt hierzu einen sogenannten `chroot`-Systemaufruf mit einem Verzeichnisnamen, wie z. B. `/usr/local/ftp`, als Argument aus (als Pfad wird das Home-Verzeichnis des Benutzers `ftp` verwendet). Der Effekt dieses Aufrufs ist der, daß der anonyme FTP-Benutzer anschließend das im `chroot`-Aufruf angegebene Verzeichnis als sein `/`-Verzeichnis sieht. Führt der FTP-Benutzer beispielsweise ein `cd /`-Kommando aus, befindet er sich aus seiner Sicht im Verzeichnis `/`. Aus Sicht des Servers befindet er sich jedoch nicht in dessen Wurzelverzeichnis, sondern im Verzeichnis `/usr/local/ftp`. Durch diese Maßnahme kann garantiert werden, daß ein anonym Benutzer keine Möglichkeit hat, an die kritischen Systemdateien des FTP-Servers, wie z. B. `/etc/passwd`, zu gelangen.

Damit der FTP-Benutzer die Möglichkeit hat, Verzeichnisse aufzulisten und evtl. Dateien zu komprimieren, müssen vom Systemadministrator einige Dateien in das FTP-Verzeichnis kopiert werden. Hierzu zählt insbesondere das Programm `ls`, u. U. auch die Programme `tar` und `compress`, die alle in das Verzeichnis `/usr/local/ftp/bin` kopiert werden sollten. Aus Sicht des anonymen FTP-Benutzers stehen die Programme also unter dem Verzeichnis `/bin`. Bezüglich der Programme muß eine Besonderheit beachtet werden: Normalerweise laden Programme wie z. B. `ls` beim Start Programmcode aus verschiedenen Bibliotheken hinzu (siehe Ausgabe von `ldd /bin/ls`). Das Problem besteht nun darin, daß bei Ausführung eines Programms durch einen anonymen FTP-Benutzer keine

dieser System-Bibliotheken geladen werden kann, da diese ja außerhalb des FTP-Basisverzeichnisses liegen, das das Wurzelverzeichnis für den Benutzer darstellt. Die Lösung des Problems liegt darin, daß nicht die üblichen Programmversion verwendet werden, sondern solche, die bei der Übersetzung statisch gebunden wurden und daher keinen Programmcode aus Bibliotheken laden müssen.

Darüber hinaus muß eine minimale Datei `/usr/local/ftp/etc/passwd` mit Einträgen für `root` und `ftp` erstellt werden. Für den `ftp`-User sollte eine sonst nicht verwendete Gruppe ausgesucht werden, die in der Datei `/usr/local/ftp/etc/group` eingetragen wird. Die `passwd`-Datei sollte keine Paßwörter, sondern einen `*` an deren Stelle enthalten. Die Dateien und Verzeichnisse sollten dem Benutzer `root` gehören und die Rechte `111` haben. Der Nutzen der beiden Dateien liegt darin, daß der FTP-Benutzer für die Dateien, die er laden kann, einen Owner-Namen und keine Owner-UID (User-Id) sieht.

Ein weiteres Verzeichnis, das angelegt werden sollte, ist `/usr/local/ftp/pub` (Owner `root`, Rechte `555`). In dieses Verzeichnis sollten alle Dateien kopiert werden, die einem anonymen Benutzer zum Download zur Verfügung stehen sollen. Ist auch die Möglichkeit des Uploads, also des Dateitransfers vom Client zum Server beabsichtigt, sollte ein weiteres Verzeichnis `/usr/local/ftp/incoming` erstellt werden, das für diesen Zweck benutzt werden kann (Owner `root`, Rechte `733`).

Normalerweise liegt einer Linux-Distribution ein bereits vorbereitetes FTP-Serverpaket bei, das die beschriebene Verzeichnishierarchie mit statisch gebundenen Programmen enthält. Bei SuSE Linux heißt das Paket `ftplib`. Sollte der verwendeten Distribution kein solches Paket beiliegen, kann es einfach vom FTP-Server `ftp.suse.com` heruntergeladen werden.

4.8.0.2 Installation und Konfiguration

Der erste Schritt zur Einrichtung eines FTP-Servers mit Hilfe von WUFTP besteht in der Installation der Software selbst. Das WUFTP-Paket kann entweder von dessen Homepage <http://www.landfield.com/wu-ftp/> als fertig übersetztes Binary oder im Quelltext bezogen werden. Wurde die Quelltext-Variante gewählt, sollte die darin enthaltene Datei `INSTALL` beachtet werden, die wichtige Hinweise zur Übersetzung des Programms enthält.

Im Anschluß an die Installation sollte überprüft werden, ob es einen Benutzer `ftp` in der Datei `/etc/passwd` gibt. Für diese Kennung sollte in der `passwd`-Datei bzw. in der `/etc/shadow`-Datei das Zeichen `*` als Passwort (2. Feld) eingetragen sein. Die Einträge in `/etc/passwd`, `/etc/shadow` und `/etc/group` könnten wie folgt aussehen:

```
# /etc/passwd:
# FTP-Basisverzeichnis ist /usr/local/ftp:
```

Kapitel 5

Audiokommunikation

Die Computer-Welt wird zunehmend multimedialer. Neben der einfachen Datenverarbeitung müssen Systeme wie Linux auch diesen Bereich abdecken. Niemand möchte heutzutage auf das Abspielen einer CD auf seinem Rechner, die Wiedergabe von MPEG-Audio-Dateien oder z. B. die Realisierung eines Anrufbeantworters mit Hilfe des Rechners mehr verzichten.

Auch Linux trägt diesem Trend in wachsendem Maße Rechnung. So stellt z. B. der Betrieb einer Soundkarte und den darauf aufbauenden Anwendungen unter Linux keine Schwierigkeit mehr dar. In diesem Kapitel werden einige der bestehenden Möglichkeiten zur Nutzung der Audiofähigkeiten unter Linux beschrieben. Hierzu gehört insbesondere die Realisierung eines *Anrufbeantworters* (sowohl für analoge als auch für ISDN-Anschlüsse), die in den folgenden Abschnitten dargestellt wird.

5.1 Grundlagen

In diesem Abschnitt sollen zum besseren Verständnis der im folgenden beschriebenen Audio-Anwendungen die notwendigen Grundlagen vorgestellt werden. Dazu gehört z. B. das Wissen, wie Töne überhaupt digitalisiert werden können und welche wichtigen Dateiformate zur Speicherung von Audiodaten existieren.

5.1.1 Digitalisierung von Audiodaten

Töne sind analoge Signale, die aus einer oder mehreren sich überlagernden Wellen bestehen. In Abbildung 5.1 auf der nächsten Seite ist ein solches Signal graphisch dargestellt. Das dort abgebildete sinusförmige Signal verändert im Laufe der Zeit (horizontale Achse) seine Amplitude, also die Lautstärke (vertikale Achse).

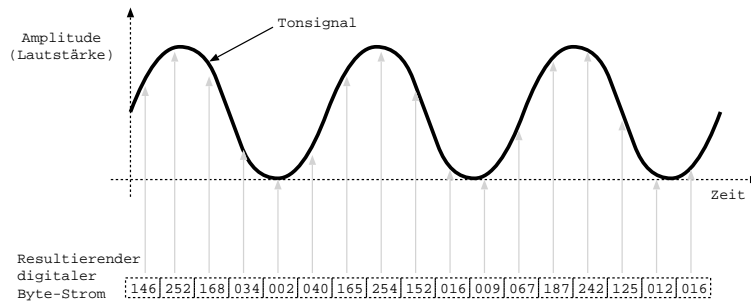


Abbildung 5.1: Digitalisierung von Audiodaten

5.1.1.1 Digitalisierung

Um aus diesem analogen Signal digitale Daten zu gewinnen, aus denen sich das ursprüngliche Signal möglichst genau wieder erzeugen läßt, wird ein übliches Verfahren angewandt, das die Amplitude des Signals in sehr kurzen Zeitabständen mißt und mit Hilfe eines Analog/Digital-Wandlers in einen digitalen Wert umwandelt, der die Lautstärke des Signals zum gemessenen Zeitpunkt darstellt. Dieses Verfahren wird als PCM (Pulse Code Modulation) bezeichnet. In oben stehender Abbildung ist die Messung der Amplitude des Signals durch Pfeile angedeutet, an deren unterem Ende der jeweils ermittelte Amplitudenwert steht, der als Zahl dargestellt ist. Die Ermittlung der Werte wird auch als Sampling bezeichnet. Ist die Zahl der Messungen (Samples) pro Sekunde ausreichend hoch, wird erreicht, daß das ursprüngliche Signal möglichst genau wiedergegeben werden kann, wobei das menschliche Ohr nicht in der Lage ist, die Lücken zwischen den gemessenen Werten zu hören. Die Sampling-Rate wird oftmals in Hertz (Abkürzung „Hz“) angegeben, obwohl die eigentlich korrekte Bezeichnung Sample/Sekunde lautet. Eine Sample Rate von 8000 Hz (= 8 KHz) besagt z. B., daß ein Signal 8000mal in der Sekunde gemessen wird. Bei jeder Messung ist das Ergebnis je nach Verfahren eine 8- oder 16-Bit große Zahl. Die sich aus der Digitalisierung eines Signals ergebende Datenrate hängt sowohl von der Sampling-Rate, als auch von der Zahl der zu digitalisierenden Kanäle (1=mono, 2=stereo) und von der Zahl der Bits je Sample (8 oder 16) ab. Soll ein Stereo-Signal zu 8-Bit großen Werten digitalisiert werden, müssen z. B. zwei Audio-Kanäle nebeneinander gemessen werden, und das Ergebnis je Sample besteht aus zwei 8-Bit großen Zahlen, wodurch eine Bitrate von $8 \cdot 2 \cdot 8000$ Bit (16 kByte) je Sekunde erzeugt wird. Die Qualität des digitalisierten Signals bei dieser Rate entspricht der des Telefons. Um eine bessere Audio-Qualität zu erzielen, wird die Sampling-Rate erhöht, wodurch sich natürlich auch das entstehende Datenvolumen vergrößert. Darüber hinaus kann zur Verbesserung der Qualität des gespeicherten Signals die Bit-Breite (zur Speicherung der Signal-Amplitude) eines Samples von 8 auf 16 Bit

erhöht werden. Übliche Sampling Raten sind 8 KHz (A-law und U-law Standard aus der Telefonie), 11 KHz (ein viertel der CD-Sampling-Rate), 22 KHz (halbe CD-Sampling-Rate) und die Sampling-Rate von 44100 Hz, die für CDs verwendet wird. Bei einem Stereosignal und fallen bei dieser Rate also $16 \text{ Bit} \cdot 2 \text{ Kanäle} \cdot 44100 = 1411200 \text{ Bit/Sec} = 176400 \text{ Bytes/sec}$ als Datenrate an. Auf eine Audio-CD, auf der ca. 650 MB gespeichert werden können, passen daher ca. 65 Minuten Audiodaten.

5.1.2 Komprimierung von Audiodaten

Wie in den vorangegangenen Abschnitten bereits dargestellt wurde, fallen je nach der verwendeten Sampling-Rate und damit der Qualität eines digitalen Audio-Signals nicht unerhebliche Datenraten an. Daher wurden Verfahren entwickelt, die durch Kompression oder geschicktere Codierung der Audio-Signale zu einer Verringerung der anfallenden Daten führen sollen.

5.1.2.1 A-Law und U-Law

Das erste dieser Verfahren besteht aus der sogenannten *A-Law-* bzw. *U-Law-Codierung*, durch die eine Komprimierung von 14 auf 8 Bit möglich wird. Die resultierende Bitrate bei dieser Codierung beträgt 64 kBit/sec. Diese Verfahren sind im ITU-T-G.711-Standard beschrieben und werden im wesentlichen in der Telefonie-Technik eingesetzt. Beide Verfahren sind nicht zueinander kompatibel. Der A-Law-Standard ist in Europa verbreitet, während in den USA die U-Law (genau μ -Law)-Codierung verwendet wird.

5.1.2.2 ADPCM

Die *ADPCM-Codierung* (*Adaptive Differential Pulse Code Modulation*) ist eine Verfeinerung der PCM-Codierung, bei der nicht der absolute Wert der Samples, sondern die Differenz zwischen Samples codiert ist. Das ADPCM-Verfahren ist in den Standards der ITU (ehemals CCITT) G.721, G.723 und ITU-T G.726 beschrieben. Dieses Verfahren ermöglicht die Digitalisierung von Audiodaten mit Sample-Raten von 16, 24, 32 und 40 kBit/sec. Die Größen der Samples sind je nach Sample-Rate 2, 3, 4 oder 5 Bit.

Neben dem beschriebenen ADPCM-Standard existieren weitere ADPCM-Verfahren, wovon insbesondere die Microsoft ADPCM-Codierung durch die Einführung des WAV-Dateiformats (in dem die ADPCM-Codierung verwendet wird) Bedeutung erlangt hat. Diese Codierung ist nicht kompatibel zu dem oben beschrieben offiziellen ITU-Standard. Das Microsoft-Verfahren ermöglicht die Komprimierung von 16 Bit Audiodaten in 4 Bit.

5.1.2.3 GSM

Der *GSM-Standard* beschreibt ein ursprünglich für die Komprimierung von Sprache im Bereich Mobiler Kommunikation entwickeltes Verfahren zur Datenreduktion. Es stellt eine Variante der LPC (Linear Prediction Coder) Codierung mit der Bezeichnung RPE-LPC (Regular Pulse Excited - Linear Prediction Coder) dar. Mit Hilfe dieses Verfahrens können 160 13-Bit große Samples zu 33 Bytes komprimiert werden, was einer Komprimierung von ca. 1 : 7,5 entspricht.

5.1.2.4 MPEG

MPEG (Motion Pictures Expert Group) ist ein Standard der ISO zur standardisierten Komprimierung von Video- und Audiodaten. Es gibt den MPEG-1-Standard (gedacht für Video-CD), den MPEG-2-Standard (Verbesserung des MPEG-1-Standards für höhere Datenraten, z. B. für DVD) und den MPEG-4-Standard.

Zur Komprimierung von Audiodaten ist das zur Zeit wohl beste Verfahren das durch den MPEG-1 Layer 3 oft einfach als „MP3“ bezeichnete Verfahren. Es ermöglicht eine wesentliche Komprimierung (ca. Faktor 12) der Audiodaten, indem für den Menschen nicht wahrnehmbare Anteile des Audio-Signals weggelassen werden. Dieser Standard sollte nicht mit dem nicht mehr existierenden MPEG-3-Standard verwechselt werden, der in den MPEG-2-Standard aufgenommen wurde.

5.1.2.5 Ogg Vorbis

Ähnlich wie MPEG ist auch Ogg Vorbis ein Verfahren zur Komprimierung von Audio-Daten. Der wesentliche Unterschied liegt darin, daß Ogg Vorbis ein Verfahren darstellt, das frei von Patenten ist. Das Ziel des Entwicklers (www.vorbis.com) ist letztlich die Erstellung eines vollkommen offenen Multimedia-Systems. Die Verwendung eines Verfahrens, das frei von Lizenzrechten und Patentgebühren ist, könnte helfen, die Preise von z. B. MP3-Playern, die dann vielleicht besser Ogg Vorbis-Player heißen sollten, zu reduzieren. Darüber hinaus ist das Fehlen von Patenten insbesondere für Open Source wichtig, da offene Implementierungen von Ogg Vorbis nicht gegen irgendwelche Lizenzrechte verstoßen. Dateien, die im OggVorbis-Format komprimiert sind, tragen normalerweise die Endung `.ogg`.

5.1.3 Wichtige Audio-Dateiformate

Zur Speicherung, Verarbeitung und Übertragung (z. B. über das Internet) von digitalen Audiodaten ist es erforderlich, diese in Dateien zu speichern. Im Laufe der Zeit haben sich zu diesem Zweck zahlreiche Dateiformate entwickelt, die über-

wiegend zunächst auf bestimmten Rechnersystemen zum Einsatz kamen, heute jedoch auf vielen unterschiedlichen Systemen verwendet werden können.

Die heute existierenden Dateiformate lassen sich grob in solche einteilen, die selbstbeschreibend sind, d. h., die neben den eigentlichen Audiodaten zusätzliche Metainformationen über das verwendete Format enthalten und evtl. sogar weitere Informationen, wie etwa eine lesbare Beschreibung der enthaltenen Audiodaten beinhalten. Die zweite Gruppe sind die Formate ohne Metainformationen, die lediglich die Audiodaten selbst enthalten, wobei lediglich aufgrund der Dateierdung darauf geschlossen werden kann, wie die Daten codiert sind.

5.1.3.1 Selbstbeschreibende Audio-Dateiformate

Die wichtigsten Dateiformate, die zu dieser Gruppe gehören, sind in Tabelle 5.1 zusammengestellt.

Tabelle 5.1: Audio-Datei-Formate mit Metainformationen

Name	Extender	Beschreibung
Au	.au	Von SUN Microsystems entwickelte AU-Dateien, die normalerweise U-Law-codierte Daten bei einer Sample-Rate von 8000 Hz enthalten. Das Dateiformat enthält Informationen über die verwendete Sample-Rate, die Zahl der Kanäle und über das verwendete Encoding (z. B. A-Law oder U-Law).
AIFF	.aif, .aiff	Das Audio Interchange File Format. Ein auf Apple II- und SGI-Rechnern verwendetes Dateiformat, das eine Codierung von Audiodaten in 8-Bit-Mono oder -Stereo ermöglicht. AIFF-Dateien sind nicht komprimiert. Zu diesem Zweck existiert ein weiteres Format mit der Bezeichnung AIFF-C oder AIFC, das Kompressionsfaktoren von ca. 1 : 6 ermöglicht. Das Dateiformat beschreibt Daten wie die Sampling-Rate, die Zahl der Kanäle, die Bit-Breite der Sample und weiteres mehr.
MP3	.mp3	Ein standardisiertes Verfahren zur Komprimierung von Audiodaten im Rahmen des MPEG-1-Video/Audio-Komprimierungsstandards. Die MP3-Komprimierung erlaubt Kompressionsfaktoren von bis zu 1 : 12. Im Dateiformat ist die Zahl der Kanäle, die Sampling-Rate, die Bit-Rate sowie weitere Informationen wie die Länge und weitere Informations-Tags (z. B. Interpret ...) codiert.

Tabelle 5.1 – Fortsetzung

Name	Extender	Beschreibung
Ogg	.ogg	Das frei Ogg-Vorbis-Format zur Komprimierung von Audio-Dateien. Die erreichte Qualität und der Kompressionsfaktor sind dem von MP3-Dateien leicht überlegen. Der Hauptvorteil dieses offenen Standards ist die Tatsache, daß keine Patente existieren, die eine Implementierung von z. B. Ogg-Encodern rechtlich brisant machen könnten, wie dies bei dem patentgeschützten MP3-Verfahren der Fall ist.
WAVE	.wav	Das Microsoft WAVE-Dateiformat, das auf dem RIFF-Format einer Abwandlung des IFF (Interchange File Format) beruht. Das Dateiformat beschreibt die Sample-Rate, die Bitbreite eines Sample, die Zahl der Kanäle, die verwendete Codierung (z. B. PCM, U-Law, A-Law, MS ADPCM, etc.).
—	.voc	Das VOC-Format ist ein Soundblaster-spezifisches Format. Es enthält Informationen über die Sample-Rate und die Zahl der Kanäle. Mit Hilfe des Formats können Stille-Pausen im Audiosignal dargestellt werden. Darüber hinaus ist auch die Darstellung von Wiederholungen (Loops) möglich.
PVF	.pvf	Spezielles Portable Voice Format, das für das Programm <code>vgetty</code> zum Betrieb eines Anrufbeantworters verwendet wird.
VBOX	.msg	Spezielles Format das von dem ISDN-Anrufbeantworterprogramm <code>vbox</code> zur Speicherung von (Audio-)Nachrichten verwendet wird. Die Datei enthält neben den Codierungsinformationen weitere Informationen über die Nachricht, Telefonnummer etc.; siehe auch Abschnitt 5.4.2 auf Seite 470.

5.1.3.2 Audio-Dateiformate ohne Meta-Information

Neben den Dateiformaten, die ihren Inhalt mit Hilfe von Meta-Informationen beschreiben, existieren auch eine ganze Reihe weiterer Dateiformate, die lediglich die Daten enthalten. Eine Beschreibung über die Art der Codierung, der Zahl der Kanäle oder der Bitbreite der einzelnen Samples fehlt. Eine Unterscheidung der Formate ist hier nur aufgrund der Dateiendung möglich. Enthält eine Datei doch ein anderes Format als das durch die Endung beschriebene, wird z. B. das Abspie-

len dieser Datei bestenfalls Geräusche erzeugen, jedoch nicht das ursprünglich aufgenommene Signal wiedergeben können. Einige Beispiele für Audio-Dateien ohne Metainformationen sind in Tabelle 5.2 dargestellt.

5.1.4 Bearbeiten/Konvertieren von Audiodaten

Die Vielzahl an existierenden Sound-Formaten macht es oftmals erforderlich, Formate zu konvertieren, da bestimmte Anwendungen, etwa ein elektronischer Anrufbeantworter, nur mit bestimmten Formaten arbeiten kann. Zu diesem Zweck existieren verschiedene Konvertierungsprogramme, die Audiodaten von einem in ein anderes Format überführen können. Das Programm unter Linux mit den vielfältigsten Möglichkeiten zu diesem Zweck ist `sox` und wird im folgenden beschrieben. Für spezielle Formate, wie z. B. das von `vgetty` (aus dem `mgetty` und `sendfax`-Paket) verwendete `pvf`-Format oder das `Vbox`-Format das von `vbox-getty` (ISDN-Anrufbeantworter) verwendet wird, existieren separate Konverter, die eine Anbindung an gängige Formate wie `.au` oder `.wav` ermöglichen.

5.1.4.1 Soundkonvertierung mit `sox`

Das Programm `sox` ist ein sehr leistungsfähiger Konverter für eine Vielzahl von unterschiedlichen Sound-Formaten. Darüber hinaus ermöglicht `sox` ebenfalls die Verwendung von Effekten, z. B. Hall oder das Setzen eines Hoch- oder Tiefpaß-Filters, um bestimmte Frequenzspektren aus dem Signal herauszufiltern, die aber hier nicht weiter besprochen werden sollen. Mehr Informationen finden sich in den Manual-Seiten (Manpage) zu `sox`.

Der Preis für diese Vielfalt an Möglichkeiten ist die Komplexität des Werkzeugs. Das Programm `sox` verfügt über eine verwirrende Zahl an Optionen. Darüber hinaus ist die Syntax der Kommandozeile komplex und nicht immer direkt ver-

Tabelle 5.2: Audio-Datei-Formate ohne Metainformationen

Name	Extender	Beschreibung
CD-R	<code>.cdr</code>	Dateien zum Mastering von CDs. Die Datei enthält in der Regel Daten für ein Stereosignal mit einer Sample-Rate von 44 kHz. Die Daten sind uncodiert und vorzeichenlos.
—	<code>.snd</code>	Ein auf Macintosh-Rechnern und PCs verwendetes Format, mit variabler Sampling-Rate, einem Kanal und 8-Bit vorzeichenlosen Daten.
—	<code>.ul</code> , <code>.al</code>	Dateien mit einem 8-kHz-Audio-Signal in U-Law- bzw. A-Law-Codierung.

Kapitel 6

Videokommunikation

Das Medium Fernsehen hat schon seit geraumer Zeit fast alle Lebensbereiche durchdrungen. Bei einer Verbreitung von ca. 99% über alle Haushalte, ist es letztlich nur konsequent, wenn auch der PC als Fernseher verwendet wird. Hierzu existieren seit einiger Zeit spezielle TV-Karten, oftmals in Verbindung mit einem zusätzlichem UKW-Tuner, die den Empfang von *Fernseh-* und *Radioprogrammen* möglich machen. Eine weitere Gruppe von Karten ermöglicht durch einen entsprechenden Tuner (zusammen mit der entsprechenden Sat-Antenne) das Ansehen von *Satelliten-Fernsehen*, wozu bei den anderen Karten ein handelsüblicher Satelliten-Receiver erforderlich wäre. Kabel oder terrestrische ausgestrahlte Programme sind mit diesen Karten jedoch nicht zu empfangen.

Mit Hilfe der Multimedia-Karten ist es neben dem Empfang von Fernseh- und ggf. Radiosendern auch möglich, weitere Dienste zu nutzen. Hierzu gehört z. B. die Möglichkeit, *Videotext* zu empfangen, und darüber hinaus auch die Decodierung von sogenannten *Intercast-Programmen*, die in Deutschland zur Zeit vom ZDF und DSF ausgestrahlt werden. Hierbei handelt es sich um Dateien, die neben dem Fernsehsignal übertragen werden, wie z. B. HTML-Seiten, Bilder oder Audio-Dateien, die nach der Decodierung, z. B. mit Hilfe eines Web-Browsers betrachtet werden können.

Die meisten Karten, basieren auf einem Chip der Firma Brooktree (heute Conexant), der als BTTV 848, BTTV 849 oder, für die neueren Varianten, als BTTV 878 und BTTV 879 bezeichnet wird. Obwohl inzwischen auch viele andere Hardware verwendet werden kann, ist die Unterstützung für Karten mit diesen Chipsatz immer noch am besten und unproblematischsten. Der Grund für diese Tatsache liegt in der historischen Entwicklung, da der erste Treiber und die ersten Anwendungen für eine solche BTTV-848-basierte Multimedia-Karte geschrieben wurden, aus dem sich letztlich die heute existierende Struktur entwickelt hat.

Neben der Gruppe von Fernseh-fähigen Karten existieren weitere, im Vergleich zu den Fernseh-Karten meist sehr preiswerte Varianten, die lediglich den Empfang von UKW-Radiosendern ermöglichen.

6.1 Das “Video for Linux”-Projekt

Der erste für Linux entwickelte Treiber für einen Multimedia-Karte diente der Steuerung eines BTTV-848-Chips. Neben dem eigentlichen Treiber wurden auch Anwendungen entwickelt, die über den Treiber auf spezielle Eigenschaften *dieser* Karte zugreifen konnten. Das Problem mit diesem ersten Ansatz war die Tatsache, das praktisch für jede weitere Multimedia-Karte, die nicht auf dem gleichen Chip-Satz beruht, ein neuer Treiber geschrieben als auch eine Portierung der Applikationen (etwa eine Fernsehapplikation) vorgenommen werden mußte. Mit der steigenden Zahl von unterschiedlichen Karten war dieser Ansatz nicht mehr praktikabel.

Ziel des *Video for Linux*- (kurz →*V4L*-) Projekts ist die Bereitstellung einer Architektur, mit der es möglich ist, Anwendungen unabhängig von der verwendeten Hardware einer Multimedia-Karte entwickeln und verwenden zu können. Zu diesem Zweck wurde eine →*API* (*Application Programming Interface*) entwickelt, auf die Entwickler von Multimedia-Anwendungen unabhängig von der verwendeten Karte in immer gleicher Weise zugreifen können. Dadurch können einmal geschriebene Multimedia-Anwendungen letztlich zusammen mit beliebigen Karten verwendet werden. Voraussetzung ist lediglich, daß es einen (*V4L*-) Treiber für die jeweilige Karte gibt, der die in der API geforderte Funktionalität für die jeweilige Karte bereitstellt. Eine Beschreibung dieser API ist unter der Homepage des *V4L*-Projekts unter der URL <http://roadrunner.swansea.uk.linux.org/v41.shtml> verfügbar. Die meisten heute für Linux existierenden Treiber sind *V4L*-fähig, d. h., sie stellen Funktionen der oben beschriebenen API bereit, wodurch zahlreiche Anwendungen genutzt werden können. Daher sollte beim Kauf einer Karte berücksichtigt werden, ob ein *V4L*-Treiber für diese Karte existiert. Genauere Informationen hierzu stehen in folgendem Abschnitt.

Da sich im Laufe der Zeit herausgestellt hat, daß auch die *V4L*-API noch immer zuviel Ballast aus dem ursprünglichen BTTV-Treiber enthält, und um mehr Flexibilität und mehr Support für weitere Geräte zu ermöglichen, wird eine verbesserte API entwickelt, die als →*V4L2* (*Video for Linux Two*) bezeichnet wird. Informationen zu diesem Projekt sind unter der URL <http://www.thedirks.org/v412/> verfügbar.

6.2 Hardware

Basis für die Nutzung von Fernsehen und Radio unter Linux ist eine entsprechende *Multimedia-Karte*. Einfache Radio-Karten werden meist als preiswerte ISA-Karten angeboten. Bei den heutigen Fernseh- oder den kombinierten Fernseh-/Radio-/Videotext-Karten handelt es sich in der Regel immer um PCI-Karten. Neben der Möglichkeit Fernsehen zu sehen, bieten diese Karten oftmals weitere Anschlußmöglichkeiten, wie z. B. einen S-Video-Eingang.

Die TV-Karten enthalten als Basis alle einen HF-(High Frequency-)Tuner sowie einen Video Grabber Chip (z. B. BTTV 848). Mit Hilfe dieses Chips wird das analoge Fernsehsignal (das FBAS = Farbbild-Austast-Synchron-Signal) in ein digitales → YUV-Signal umgewandelt. Die gewonnene Bildinformation wird in der Regel direkt zu der Graphikkarte zur Darstellung übertragen.

Hierbei existieren wiederum zwei verschiedene Verfahren, die als Primary- und als Overlay-Verfahren bezeichnet werden. Das von einer Graphikkarte unabhängige Primary-Verfahren überläßt die Berechnung der Bildauflösung der TV-Karte, die das Bild an die Graphikkarte zur Darstellung weiterreicht. Der Vorteil liegt darin, das dieser Modus mit jeder Graphik-Karte arbeitet, das Bild jedoch in der maximalen PAL-Auflösung von 768x576 Punkten dargestellt werden kann. Beim Overlay-Modus wird die Berechnung der Bildinformation und der Auflösung von der Graphikkarte übernommen, wodurch das Bild frei skalierbar ist und bei beliebigen Auflösungen als Vollbild dargestellt werden kann. Der Nachteil ist, das dieser Modus nicht von jeder Karte unterstützt wird.

Neben dem Tuner und dem Grabber Chip kommt häufig noch ein FM-Tuner zum Empfang von Radio-Sendern zum Einsatz als auch ein spezieller Sound-Chip, der zur Wiedergabe von Stereo-Klang verwendet wird.

6.2.1 Unterstützte Hardware

Vor der Kaufentscheidung zugunsten einer bestimmten Karte sollte der Support dieser Karte unter Linux geprüft werden, damit die Karte möglichst problemlos verwendet werden kann. Sinnvoll ist auf jeden Fall darauf zu achten, daß ein V4L-Treiber für die entsprechende Karte existiert, damit anschließend möglichst viele Anwendungen mit dieser Karte genutzt werden können. Inzwischen unterstützt Linux eine große Zahl an unterschiedlichen Karten und Chipsätzen. Da die Liste der unterstützten Karten sich in ständiger Veränderung befindet, ist es nicht sinnvoll, an dieser Stelle eine feste Liste anzugeben. Grundsätzlich kann jedoch gesagt werden, daß die beste Unterstützung für Karten existiert, die auf dem BTTV 848 (und 849/878/879)-Chip basieren. Eine Liste mit der aktuell unterstützten Hardware kann unter folgenden URLs eingesehen werden:

1. In der SuSE-Hardwaredatenbank:
<http://hardwaredb.suse.de/>
2. Auf den Multimedia4Linux-Seiten:
<http://www.multimedia4linux.de/videohardware.html>
<http://www.multimedia4linux.de/videomodule.html>
3. Einige konkrete Produktempfehlungen sind auf den „Video for Linux Resources“-Seiten verfügbar:
<http://www.exploits.org/v4l/>

6.2.2 Automatische Konfiguration einer TV-Karte

Mit Hilfe von `yast2` ist die Konfiguration einer TV-Karte heute kein Thema mehr. Sie erfolgt in der Regel schon bei der Installation von SuSE Linux. Falls die Karte erst nach der Installation eingebaut wurde, kann die Konfiguration leicht mit Hilfe von `yast2` ausgeführt werden.

In Abbildung 6.1 ist ein Schritt der Konfiguration einer TV-Karte mit `yast2` dargestellt. Der Aufruf des TV-Karten-Konfigurationsmenüs erfolgt aus dem `yast2`-Menüpunkt `Hardware` heraus.

Wer mehr über den Konfigurationsvorgang und der dahinter stehenden Technik erfahren möchte, oder falls in einem seltenen Fall einmal die automatische Konfiguration nicht richtig funktionieren sollte, kann den folgenden Abschnitt lesen, in dem die manuelle Konfiguration einer TV-Karte auf Basis des weit verbreiteten BTTV-Chips beschrieben wird.

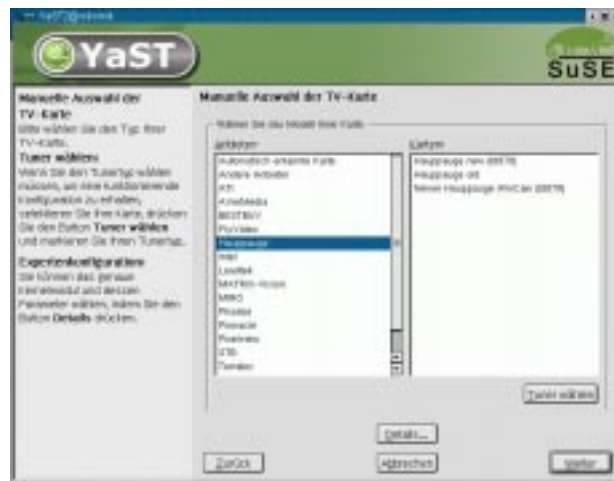


Abbildung 6.1: Die `yast2` TV-Karten-Konfiguration

6.2.3 Konfiguration einer BTTV-TV/Radio-Karte

Im folgenden soll die Konfiguration einer der am weitest verbreiteten TV/Radio-Karten, die auf dem BTTV 878-Chipsatz beruht, beschrieben werden. Die für die folgende Beschreibung zugrunde liegende Karte ist eine Hauppauge Win/TV Radio. Die Beschreibung ist jedoch bis auf wenige Parameter, die den genauen Typ einer Karte angeben, identisch mit anderen BTTV-basierten Karten. Eine Liste solcher Karten kann unter einer der oben genannten URLs eingesehen werden. Für die Installation einer Karte, die auf einem anderen Chip-Satz, z. B. auf einem Zoran-Chip, beruht, kann unter der URL <http://www.multimedia4linux.de/videomodule.html> ein Link zu einem entsprechenden Treiber gefunden werden. Für die Installation sollte das README des Treibers gelesen werden.

Für den Betrieb einer TV-Karte sollte auf jeden Fall ein aktueller Kernel verwendet werden. Die Kernel-Module für verschiedene TV-Karten wurden im Laufe der Entwicklung in den 2.4er-Kernel integriert. Aber auch für ältere Kernel können noch Treiber zur Verfügung stehen. Wer nicht sicher ist, kann das Vorhandensein der BTTV-Module selbst testen. Im Verzeichnis `/lib/modules/`uname -r`/` sollten in diesem Fall die Dateien `videodev.o`, `i2c.o`, `bttv.o`, `tuner.o`, `misp3400.o` existieren.

Die Konfiguration ist grundsätzlich sehr einfach, da sie lediglich aus dem Laden von Kernel-Modulen besteht, denen mit Hilfe von Parametern der genaue Typ der Karte bzw. der einzelner Chips angegeben werden kann. Die verschiedenen Parameter einer bestimmten Karte, die für die jeweiligen Kernel-Module verwendet werden sollten, können aus dem Video-TV HOWTO in Abschnitt 3 unter der URL

<http://www.multimedia4linux.de/howto/DE-Video-TV-HOWTO.html> entnommen werden.

6.2.3.1 Download des BTTV-Treibers

Unter Verwendung eines aktuellen 2.4er oder des etwas älteren 2.2er Kernels, ist der Treiber schon Bestandteil des Kernel. Ein extra Download ist in diesem Fall nicht erforderlich.

Falls ein älterer Kernel verwendet wird, bei dem die oben genannten Module noch nicht integriert sind, können diese mit Hilfe der Treiber-Quelldateien erstellt werden (siehe README des Treibers). Darüber hinaus enthält der Quellcode auch ein Skript, `update`, mit dem es auf einfache Weise möglich ist, alle benötigten Kernel-Module in einem Aufruf zu entladen und anschließend neu zu laden. Dies ist sehr nützlich, da insbesondere im Problemfall unter Umständen mit verschiedenen Parametern der benötigten Module experimentiert werden muß.

<http://www.studiolproductions.com/Articles/411samp.htm>
http://www.joemaller.com/fcp/fxscript_yuv_color.shtml

6.8.5.2 Video-Player und Recorder

- ❑ **Der Videorecorder dvr**
<http://dvr.sourceforge.net/>
- ❑ **Der Video-Encoder/Streamer ffmpeg**
<http://ffmpeg.sourceforge.net/>

- ❑ **Der Video-Player mplayer**
<http://www.mplayerhq.hu/homepage/>
- ❑ **Der KDE Multimedia-Player noatun**
<http://noatun.kde.org/>
- ❑ **Der Video-Player xine**
<http://xine.sourceforge.net/>
- ❑ **Der Video-Player ogle**
<http://www.dtek.chalmers.se/groups/dvd/index.shtml>
- ❑ **Die Player aktion und xanim**
<http://www.geocities.com/SiliconValley/Haven/3864/aktion.html>
<http://xanim.va.pubnix.com/home.html>
- ❑ **Der Xmovie-Player**
<http://heroines.sourceforge.net/xmovie.php3>

6.9 Video-Konferenzen und IP-Telefonie

Neben der Nutzung des Internet in Form von E-Mails und Chats haben sich in den letzten Jahren weitere Formen der Kommunikation etabliert, die auf der Nutzung des Internet basieren. In diese Kategorie fällt sicherlich die IP-Telefonie und das Abhalten von Video-Konferenzen.

Während die klassische Telefonie noch heute im wesentlichen auf die Übertragung von Sprache beschränkt ist, bietet die IP-Telefonie weitergehende Möglichkeiten. Dabei sind IP-Telefonie und Video-Konferenzen letztlich zwei Seiten derselben Medaille. Bei beiden werden im Gegensatz zur klassischen Telefonie keine Töne, also analoge Signale, sondern digitale Daten übertragen. Diese Daten können dabei sowohl digitalisierte Audio- als auch Video-Daten enthalten.

Der Vorteil der IP-Telefonie ist deren Flexibilität. Gerade für Unternehmen, die eine neue Kommunikationsinfrastruktur schaffen müssen, bietet der Aufbau eines reinen IP-Telefonienetzwerks durchaus eine Alternative zur herkömmlichen

Telefonanlage. Da heute in jedem Unternehmen eine EDV-Vernetzung aufgebaut werden muß, kann diese gleich für die Telefonie mitgenutzt werden, was den Aufbau einer zweiten Verkabelung für eine Telefonanlage erspart und so helfen kann Kosten zu sparen. Auch bei der täglichen Arbeit kann IP-basierte Telefonie das Leben erleichtern. So läßt sich ein IP-Telefon nicht nur als eigenständige Hardware implementieren, sondern auch als Software, die auf dem ohnehin genutzten PC arbeitet. Hier bieten sich dann weite Möglichkeiten der softwareseitigen Integration an.

Neben der Nutzung in Firmen bietet die IP-Telefonie auch für Privatleute eine interessante Ergänzung der Kommunikation, wenn man an die breite Verfügbarkeit des Internet bei Privathaushalten denkt. So lassen sich für den Preis des Internetzugangs Gespräche und Video-Konferenzen mit Freunden und Bekannten, auch im Ausland, sehr kostengünstig führen.

Neben den Vorteilen der IP-Telefonie gibt es natürlich auch Nachteile. Bei der klassischen Telefonie wird für ein Gespräch eine eigene Leitung zwischen den Gesprächspartnern geschaltet, die exklusiv genutzt werden kann. Bei der IP-Telefonie hingegen basiert die Übertragung der Sprache und Bilder auf dem Transport von Daten-Paketen über das Internet. Das bedeutet, daß man hier keine eigene, freie Leitung mit garantierter Übertragungsrate nutzen kann. Je nach Auslastung einzelner Strecken im Internet kann es auf dem Weg von einem zum anderen Gesprächsteilnehmer zu Verzögerungen kommen, die insbesondere bei gesprochener Sprache vom Menschen als sehr störend empfunden werden. Es darf also möglichst nicht zu einer zu hohen Verzögerung beim Transport kommen, da ansonsten keine sinnvolle Kommunikation möglich ist.

In diesem Kapitel soll zunächst eine allgemeine Übersicht über das Thema IP-Telefonie gegeben werden. Darüber hinaus wird konkret auf Anwendungen für Linux eingegangen, mit deren Hilfe IP-Telefonate und Video-Konferenzen abgehalten werden können.

6.9.1 Übersicht über verschiedene Systeme

Bevor mehr über die Struktur und den Aufbau eines IP-Telefonie/Video-Konferenzsystems gesagt wird, soll in diesem Abschnitt versucht werden, Kategorien zu bilden, in die existierende Systeme eingeordnet werden können, um auf diese Weise einen leichteren Überblick zu erhalten.

Das erste und wahrscheinlich wichtigste Unterscheidungsmerkmal ist die Unterscheidung in Systeme, die einem offenen Standard entsprechen, und solchen, die mit herstellereigenen, nicht offenen Protokollen arbeiten. Letztere Systeme müssen meist als Komplettpaket von einem Hersteller gekauft werden, wobei auch die anderen Teilnehmer über die gleichen Systeme verfügen müssen, um z. B. an der Video-Konferenz teilnehmen zu können. Auf der anderen Seite stehen die

Systeme, die nach internationalen Standards, wie z. B. H.323, arbeiten, wodurch sie in der Lage sind, mit beliebigen anderen Systemen eine Audio-Verbindung oder eine Video-Konferenz aufzubauen. Hier entfällt also die Bindung an einen bestimmten Hersteller.

Ein weiteres Merkmal ist die Art der Verbindung, die von einem Konferenzsystem verwendet wird. Hier besteht zum einen die Möglichkeit, die Verbindung über das Telefonnetz zu realisieren, da unter Nutzung von ISDN eine Verbindung mit garantierter Bandbreite zum anderen Teilnehmer zur Verfügung steht. Da ISDN jedoch lediglich $2 \cdot 64\text{KBit/sec}$ an Daten übertragen kann, verwenden solche Anlagen in der Regel z. B. drei parallel betriebene ISDN-Leitungen, um eine höhere Bandbreite, insbesondere für eine bessere Bildübertragung, zu erzielen. Auf der anderen Seite stehen die Systeme, bei denen die Ton- und Bild-Daten über ein Netzwerk, meist über ein LAN², bei Verbindungen „außer Haus“ über das Internet übertragen werden.

Schließlich kann man zwischen reinen Software- und Hardware-basierten Lösungen unterscheiden. Mit Software-Lösungen sind hierbei die Systeme gemeint, die ausschließlich durch Software realisiert sind. Im Gegensatz dazu existieren auch Lösungen, die meist als Komplettpakete eines Herstellers zusammen mit der notwendigen Hardware verkauft werden.

In diesem Kapitel werden reine Softwaresysteme, die auf dem Standard H.323 basieren, beschrieben, die unter Linux arbeiten. Daher wird im folgenden Abschnitt zunächst ein Blick auf H.323 geworfen werden.

6.9.2 Protokolle und Codecs

Zur Realisierung von IP-Telefonie als auch von Video-Konferenzen bedarf es standardisierter Protokolle, die es ermöglichen, entsprechende Anwendungen auf beliebigen zu den Standards kompatiblen Systemen miteinander „reden“ zu lassen. Dabei muß zunächst grundsätzlich zwischen zwei verschiedenen Protokolltypen unterschieden werden. Die erste Gruppe ist die Gruppe der Verbindungsprotokolle, die zweite Gruppe ist die, die man als Übertragungs- oder Benutzerprotokolle bezeichnen könnte.

Die erste Gruppe, zu der Protokolle wie H.323, Megaco oder auch SIP gehören, dient ausschließlich dem Management einer IP-Telefonie- bzw. IP-Video-Konferenzverbindung. Diese Protokolle dienen gewissermaßen dazu, die Möglichkeiten der herkömmlichen Telefonie in einem paketorientierten Netzwerk nachzubilden. Sie werden oft als Call Processing Protokolle bezeichnet. Hierzu zählen so grundlegende Dinge wie die Signalisierung eines Anrufs, der Aufbau und Abbau einer Verbindung, aber auch Dienste, wie z. B. die Aushandlung von gemeinsamen Übertragungsstandards (den Codecs) für die Ton und Bildinformation. Alle

²Ein Local Area Network, also ein lokales Netzwerk.

diese Protokolle setzen auf die zweite im folgenden beschriebene Gruppe von Protokollen auf, mit deren Hilfe die eigentliche Datenübertragung durchgeführt wird.

In die zweite Gruppe von Protokollen fällt z. B. das Standard-Protokoll TCP/IP, mit dessen Hilfe letztlich auch IP-Telefonie-Daten über das Internet übertragen werden können. TCP baut dabei auf das IP-Protokoll auf und ermöglicht fehlerfreie Verbindungen. Dies wird letztlich durch eine Fehlerkorrektur erreicht, u. a. auch auf Basis von Quittungen, die der Empfänger zum Sender zurückschickt, um den Erhalt von Daten zu bestätigen. In der IP-Telefonie ist das UDP-Protokoll, ein weiteres Protokoll, das auf IP aufsetzt, noch wichtiger als TCP. UDP arbeitet im Gegensatz zu TCP ohne das Versenden von Quittungen für erhaltene Daten. Daher kann dieses Protokoll schneller arbeiten, wobei es aber auch durchaus möglich ist, daß einzelne Datenpakete verloren gehen. Daneben existieren weitere Protokolle, wie RTP (Real Time Protocol), mit dessen Hilfe die Problematik angegangen werden kann, daß z. B. Daten für IP-Telefonie als Echtzeit-Daten mit Vorrang behandelt werden, um auf diese Weise Zeitverzögerungen beim IP-Telefonat zu verringern.

Neben den Protokollen, die für das Management der Verbindung und den Transport der Daten benötigt werden, werden noch Verfahren zur Digitalisierung von analogen Daten benötigt. Solche Verfahren werden als Codec bezeichnet und ermöglichen es, analoge Sprach- und Bilddaten in eine möglichst gut komprimierte, digitale Form umzuwandeln.

6.9.2.1 Das H.323-Protokoll

H.323 steht für eine ganze Gruppe von Protokollen mit dem Ziel, Audio-, Video-Konferenzen als auch Datenübertragungen über LANs (Local Area Networks), und ganz allgemein über paketbasierte Netzwerke zu ermöglichen. Dabei wurde beim Design von H.323 insbesondere auch berücksichtigt, daß solche Netzwerke (wie z. B. das Internet selbst) meist keine garantierte Quality of Service (QoS) besitzen, man also nicht davon ausgehen kann, daß Daten mit fest zur Verfügung stehender Bandbreite ohne Verzögerungen übertragen werden können.

H.323 ist ein Standard, der von der →ITU erstmals 1996 verabschiedet und seitdem weiterentwickelt wurde. Durch das herstellerneutrale und auch netzwerkunabhängige Design wird es möglich, daß alle H.323-kompatiblen Systeme unabhängig von dem System, auf dem sie arbeiten, und auch unabhängig von dem z. B. im lokalen Netz verwendeten Protokoll eine Verbindung miteinander aufbauen können.

Jedes H.323-kompatible System muß dabei Mindestanforderungen erfüllen. Jedes H.323-fähige Endgerät (Terminal genannt) muß beispielsweise den Audio-Codec mit der Bezeichnung G.711 (siehe unten) unterstützen. Dies ist eine Mini-

che bietet, ist beispielsweise `ohphone` eher eine IP-Telefonieanwendung ohne graphische Oberfläche. Beide sind jedoch H.323-kompatibel, so daß man mit `ohphone` auch einen `gnomemeeting`-Benutzer anrufen kann. Durch die Tatsache, daß auch Windows NetMeeting H.323-kompatibel ist, können mit allen Programmen auch Verbindungen mit Benutzern von NetMeeting aufgebaut werden.

Alle Anwendungen sind zunächst direkt in der Lage, eine zweier-Konferenz abzuwickeln, also ein Gespräch/eine Video-Konferenz zwischen genau zwei Teilnehmern. Um eine Konferenz aus drei oder mehr Teilnehmern bilden zu können, ist zusätzlich die Verwendung von `openmcu` notwendig. Diese Anwendung wird in Abschnitt 6.9.4.3 auf Seite 613 beschrieben.

6.9.4.1 Videokonferenzen mit `gnomemeeting`

Die Anwendung `gnomemeeting` stellt das in der Linux-Welt zur Zeit komfortableste und wahrscheinlich auch am weitesten verbreitete H.323-kompatible IP-Telefonie/Video-Konferenz-System dar. `gnomemeeting` wartet mit einer leicht zu bedienenden graphischen Oberfläche auf, über die auch das gesamte Setup der Anwendung erfolgt. Neben der Möglichkeit, Video-Konferenzen durchzuführen, bietet `gnomemeeting` auch die Möglichkeit, Chats durchzuführen, also Texte an den Kommunikationsteilnehmer zu senden. Als weiteres Feature bietet `gnomemeeting` Zugang zu einem ILS (Internet Locator Server), mit dessen Hilfe ein Benutzer von `gnomemeeting` und auch von Windows NetMeeting sehen kann, welche anderen Benutzer zur Zeit gerade online sind. Durch ein Klick kann dann eine Verbindung zu dem gewünschten Nutzer aufgebaut werden. Auf Wunsch registriert `gnomemeeting` den Anwender nach dem Start automatisch beim ILS-Server, so daß man für andere direkt erreichbar ist. Der ILS-Server erspart also das Suchen nach der IP-Adresse des Kommunikationspartners und bietet ein Verzeichnis der gerade aktiven `gnomemeeting`-Nutzer. Auch wenn `gnomemeeting` unter der Benutzeroberfläche Gnome entwickelt wurde, funktioniert es dennoch in gleicher Weise auch unter KDE, allerdings ist hierzu u. U. ein kleiner Trick notwendig, der weiter unten beschrieben wird.

Da `gnomemeeting` voll H.323-kompatibel ist, und das Windows Pendant Windows NetMeeting ebenfalls H.323 kompatibel ist, können problemlos Video-Konferenzen zwischen beiden Programmen durchgeführt werden.

Installation

Die Installation von `gnomemeeting` kann wie üblich durch das Übersetzen des Quellcodes erfolgen, oder durch ein bereits fertig übersetztes RPM-Paket. Sowohl der Quellcode als auch fertig übersetzte RPMs und Debian-Pakete können von der `gnomemeeting`-Homepage unter der URL <http://www.gnomemeeting>.

org bezogen werden. Nutzer von SuSE Linux haben es noch etwas einfacher, da ein fertiges RPM der SuSE Linux-Distribution beiliegt.

Neben dem eigentlichen Anwendungs-Paket werden zur Nutzung von `gnome-meeting` weitere Software-Pakete benötigt, die ebenfalls installiert werden müssen. Auf SuSE Linux werden diese Abhängigkeiten automatisch erkannt, so daß mit der Installation von `gnomemeeting` auch weitere benötigte Softwarepakete mit installiert werden.

Da `gnomeeting` eine Gnome-Applikation ist, werden zunächst die Gnome-Libraries benötigt. Wer normalerweise unter Gnome arbeitet, hat sicherlich schon alles Notwendige installiert. Wer normalerweise mit KDE arbeitet, sollte am besten einfach ein Gnome-Basis-System installieren. Neben Gnome benötigt `gnomemeeting` ebenfalls die `pwlib`, eine portable Klassenbibliothek, die auch vom OpenH323-Projekt (<http://www.OpenH323.org/>) verwendet wird. Auch die Bibliotheken vom OpenH323-Projekt selbst werden benötigt. Weiterhin wird die OpenLDAP-Bibliothek für den ILS-Support benötigt. Darüber hinaus empfiehlt sich auch die Installation der SDL-Bibliothek, wodurch `gnomemeeting` in der Lage ist, Video als Fullscreen und in einem eigenen Fenster darzustellen. Wer `gnomemeeting` selbst übersetzt und anschließend unter KDE betreiben möchte, sollte zusätzlich auch die `arts`-Entwickler-Pakete installieren, damit `gnome-meeting` anschließend unter KDE mit `artsd` umgehen kann, der unter KDE genutzt wird, um das Sound-Gerät für mehrere Anwendungen parallel nutzbar zu machen. Ansonsten kann es unter KDE vorkommen, daß bei aktiviertem `arts`-Soundserver `gnomemeeting` keinen Zugriff auf das Sound-Device hat und somit keinen Ton wiedergeben kann.

Wer nicht auf vorübersetzte Pakete zugreifen kann oder das Paket selbst übersetzen möchte, findet in der FAQ (auf der Projektseite unter der URL <http://www.gnomemeeting.org/>) von `gnomemeeting` hilfreiche Hinweise hierzu.

Als Besonderheit kann `gnomemeeting` auch mit Quicknet-Karten umgehen. Diese Karten sind spezielle Sound-Karten, die Tonsignale ohne Belastung des eigentlichen Rechners optimal zur Verwendung bei der IP-Telefonie komprimieren sollen. Auf diese Karten wird an dieser Stelle nicht weiter eingegangen. Weitere Informationen finden sich unter der URL <http://www.linuxjack.com/>.

Arbeiten mit `gnomemeeting`

Im Anschluß an die Installation kann der erste Start von `gnomemeeting` erfolgen. Wer unter Gnome arbeitet, sollte hierbei keine Probleme haben. Wer allerdings unter KDE oder mit einem anderen Windowmanager arbeitet, könnte das Phänomen erleben, daß nach dem Start scheinbar einfach nichts passiert, d. h., es erscheint kein Fenster aber auch keinerlei Fehlermeldungen. Das Problem ist in der `gnomemeeting`-FAQ beschrieben und liegt an einem Fehler im Gnome2-

System. Das Problem läßt sich allerdings leicht umgehen, indem vor dem Start der Anwendung selbst das Gnome2-Programm `bonobo-activation-empty-server` gestartet wird. Wer von diesem Problem betroffen ist und den zusätzlichen Start nicht immer von Hand ausführen möchte, kann auch ein kleines Shell-Skript verwenden, das dann anstelle von `gnomemeeting` gestartet wird:

```
#!/bin/sh
#
# Gnome-Startskript für KDE-Nutzer
#
if [ `ps -axw|grep -v grep|grep -c bonobo-activation-server` -eq 0 ]; then
    #         Bitte Pfad überprüfen !!!!
    /opt/gnome2/bin/bonobo-activation-empty-server &
    sleep 1
fi
# Bitte Pfad überprüfen !!!!
/opt/gnome2/bin/gnomemeeting
```

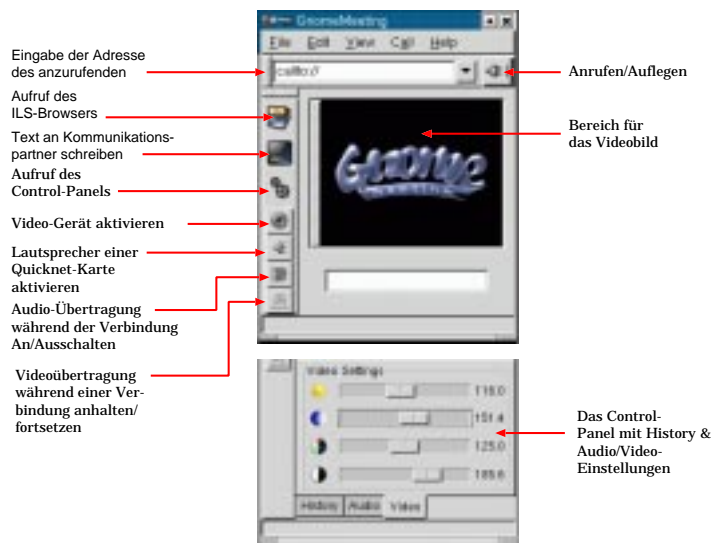
Dieses kleine Skript, was z. B. `mymeeting` genannt werden könnte, macht nichts anderes als zu überprüfen, ob der Prozeß `bonobo-activation-server` bereits läuft. Wenn nicht, wird `bonobo-activation-empty-server` gestartet und anschließend `gnomemeeting`. Wenn das Skript verwendet wird, sollte überprüft werden, ob `gnomemeeting` und `bonobo-activation-empty-server` wirklich unter `/opt/gnome2/bin` installiert sind. Ggf. müssen die Pfade im Skript angepaßt werden.

Der erste Start

Nach dem allerersten Start von `gnomemeeting` erscheint zunächst ein Assistenten-Fenster, in dem der Benutzer seine persönlichen Daten eingeben kann, also insbesondere den Namen, die E-Mail-Adresse und den Wohnort/Land. Diese Daten werden, falls der ILS-Support-Button aktiviert ist, später zum ILS-Server übertragen und sind dann für andere Nutzer sichtbar. Da die Liste des ILS-Servers auch durchsucht werden kann, sollte man die Angaben sinnvoll gestalten, wenn man gefunden werden möchte.

Darüber hinaus wird der Benutzer nach der Art seiner Internet-Anbindung gefragt. Hier kann z. B. zwischen einem 56k-Modem, ISDN oder DSL gewählt werden. Die Angabe wird zur Optimierung von Einstellungen verwendet. Die vorgenommenen Einstellungen können auch später noch jederzeit verändert werden.

Nachdem der Konfigurations-Assistent beendet wurde, erscheint das in Abbildung 6.30 auf der nächsten Seite dargestellte Fenster. Im unteren Teil der Abbildung ist zusätzlich das Control-Panel von `gnomemeeting` dargestellt, das durch den entsprechenden Button am linken Rand des Hauptfensters geöffnet und wieder geschlossen werden kann.

Abbildung 6.30: Die `gnomemeeting` Start-Fenster und Control-Panel

Aufbau einer Verbindung

Aus dem Hauptfenster heraus kann jetzt direkt ein Gesprächspartner angerufen werden. Falls das Gegenüber sich nicht im lokalen Netz befinden sollte, muß jetzt zunächst eine Verbindung zum Internet geschaffen werden. Wer auf seinem Rechner eine Firewall betreibt, sollte zuvor die Hinweise in Abschnitt 6.9.4.1 auf Seite 608 beachten.

Der Aufbau einer Verbindung erfolgt entweder durch Eingabe einer Adresse, z. B. einer `callto://`-URL in das am oberen Bildschirmrand liegende Texteingabefeld, oder mit Hilfe des ILS-Directory-Browsers, der in Abschnitt 6.9.4.1 auf der nächsten Seite beschrieben wird.

Bei der direkten Eingabe einer Adresse kann diese in unterschiedlicher Art und Weise angegeben werden. Die allgemeine Form einer Eingabe ist

```
callto://[alias]@[host:[port]]
```

oder einfach `ip-address`:

□ `ip-address`

Eine Verbindung zu einem Teilnehmer, der sich an einem Rechner mit der IP-Adresse `ip-address` befindet, kann einfach durch Eingabe der IP-Adresse, des Rechnernamens mit Domainnamen oder durch eine entsprechende `callto://`-URL, also z. B. `callto://192.168.10.5`, aufgebaut werden.

□ **Das Linux IrDA-Howto**

<http://mobilix.org/Infrared-HOWTO/Infrared-HOWTO.html>

7.3 Wireless LANs unter Linux

Der Name Wireless LAN steht für eine in der letzten Zeit immer stärkere Verbreitung findende Form der Vernetzung bei Local Area Networks, also für lokale Netzwerke. Das besondere an der Wireless Technologie liegt, wie der Name schon ausdrückt, darin, daß keine Kabel zur Vernetzung von Rechnern mehr benötigt werden. Stattdessen wird ein funkbasiertes Netzwerk aufgebaut. Drahtlose Netzwerke-Produkte sind schon eine ganze Weile auf dem Markt, jedoch ohne eine große Verbreitung erreicht zu haben. Das Problem lag bisher darin, daß es keine Standards für die drahtlose Vernetzung gab, so daß nur Produkte eines Herstellers miteinander „reden“ konnten. Dies änderte sich erst im Jahr 1997, in dem das Institut für Electrical and Electronics Engineers (IEEE) einen gemeinsamen, herstellerunabhängigen Standard für drahtlose Netzwerke unter der Bezeichnung 802.11 verabschiedete.

Die Anwendungsgebiete von WLANs, wie Wireless LANs auch bezeichnet werden, sind vielfältig. So sind sie z. B. ideal für die Vernetzung mobiler Geräte, wie Laptops oder PDAs, geeignet, die innerhalb der Reichweite des WLANs frei bewegt werden können. Auch für Situationen, in denen eine kabelgestützte Vernetzung aus baulichen oder ähnlichen Gründen nicht möglich oder zu teuer erscheint, kann ein WLAN die Lösung darstellen. Auch die Möglichkeit im privaten Bereich, z. B. vom Garten aus, im Web zu surfen und seine E-Mail zu erledigen, oder die Möglichkeit, mit Freunden oder Nachbarn einen DSL-Anschluß gemeinsam nutzen zu können, ohne dazu Löcher durch die Wände bohren zu müssen, kann ein Anwendungsgebiet für WLANs darstellen.

Schließlich werden in Zukunft immer mehr sogenannte Hot-Spots an öffentlichen Stellen, wie z. B. Flughäfen oder Cafes, Restaurants . . . , die kommerzielle Nutzung von WLANs ermöglichen, indem sie Reisenden die Möglichkeit bieten, an solchen Hot-Spots drahtlosen Zugriff auf das Internet oder auch auf die Server der eigenen Firma zu erlangen.

7.3.1 Einführung

Wireless LANs bieten eine komfortable Möglichkeit, Rechner miteinander zu vernetzen. Bevor auf die Konfiguration solcher Netze mit Linux eingegangen wird, werden zunächst einige Grundlagen zu diesem Thema dargestellt.

7.3.1.1 Technische Grundlagen

Einer der ersten verfügbaren Wireless Standards war IEEE 802.11, der bereits 1997 verabschiedet wurde. 802.11 erlaubte drahtlose Netzwerke mit einer Übertragungsrate von bis zu 2MBit/sec. Höhere Übertragungsraten wurden erst mit dem 1999 verabschiedeten Standard 802.11b verfügbar, mit dem bis zu 11MBit/sec übertragen werden können. Beide Standards arbeiten in dem sogenannten ISM (Industrial, Scientific und Medical)-Band im Frequenzbereich von 2.45 GHz. Die Funkübertragung erfolgt dabei auf einzelnen Kanälen, wobei jeder Kanal für eine leicht unterschiedliche Frequenz steht. Auf diese Weise können mehrere Sende-/Empfangseinheiten nebeneinander arbeiten, ohne sich gegenseitig zu stören. In Europa sind insgesamt 13 Kanäle verfügbar, in anderen Ländern, insbesondere in den USA sind nur 11 Kanäle erlaubt. Daher werden auch die Geräte oftmals als European- oder World-Edition vermarktet, die dann entweder über 13 oder nur 11 Sende-/Empfangskanäle verfügen.

Eine weitere Steigerung der Übertragungsrate auf bis zu 54MBit/sec verspricht der ebenfalls 1999 verabschiedete Standard 802.11a. Allerdings werden Geräte, die entsprechend 802.11a arbeiten, nicht mehr mit Geräten kompatibel sein, die mit 802.11b arbeiten, da die 54MBit-Technologie ein anderes Frequenzband im Frequenzbereich von 5 GHz verwendet. Um dennoch kompatibel zu den inzwischen sehr weit verbreiteten Geräten zu bleiben, die im 2.45-GHz-Band arbeiten, können die neuen Geräte mit zwei Sendern/Empfängern ausgestattet werden, so daß sie in beiden Frequenzbändern arbeiten können. Eine Alternative verspricht der noch in Entwicklung befindliche Standard 802.11g, mit dessen Hilfe Geräte im 2.45-GHz-Band Datenübertragungsraten von bis zu 54Mbit/sec erreichen, ohne die Kompatibilität zu den 802.11b-Geräten aufgeben zu müssen. Erreicht wird diese Leistungssteigerung durch verbesserte Modulationstechniken.

Zur Zeit werden fast ausschließlich Geräte verkauft, die nach 802.11b arbeiten, also eine maximale Datenübertragungsrate von 11Mbit/sec erreichen. 802.11b erlaubt neben der Übertragungsrate von 11Mbit weitere Level mit 5.5 und 2 und 1 Mbit/sec. Die Geräte wählen je nach Signalstärke die beste Übertragungsrate selbständig aus, falls keine festen Konfigurationsvorgaben existieren.

Um sicherzustellen, daß Geräte unterschiedlicher Hersteller nicht nur auf dem Papier miteinander kompatibel sind, sondern auch in der Praxis zusammen arbeiten, wurde eine Testsuite für 802.11b-Geräte von der Wireless Ethernet Compatibility Alliance (WECA) entwickelt, einem Zusammenschluß verschiedener Hersteller von Wireless-Produkten. Jedes Produkt, das diese Tests besteht, erhält ein WIFI (Wireless Fidelity)-Emblem. Die WIFI-Auszeichnung von Produkten soll dem Anwender die Sicherheit geben, daß alle so ausgezeichneten Geräte herstellerübergreifend untereinander kompatibel sind.

Ein wichtiger Punkt für den Aufbau und Betrieb eines Funknetzes ist natürlich die Reichweite der Funksignale. Die Reichweite hängt sowohl von der Sende-

leistung der beteiligten Geräte als auch von der örtlichen Gegebenheiten ab. Typische Wireless-Geräte arbeiten mit einer Sendeleistung von 35mW (milli Watt), einer im Vergleich z. B. zu Handys, die mit bis zu 2000 mW senden können, eher geringen Leistung. Sehr wichtig sind die örtlichen Gegebenheiten. So kann im Freien eine Reichweite von bis zu ca. 500 m erzielt werden, wo hingegen in massiv gebauten Gebäuden evtl. nur 30 m überbrückt werden können. Je massiver die Baustruktur ist, wobei Metallbeschichtungen, wie z. B. Aluminium-Folie zur Wärmeisolation, besonders abschirmend wirken, desto geringer ist die Reichweite der Wireless-Geräte.

7.3.1.2 Wireless und Linux

Linux unterstützt schon seit einiger Zeit Wireless-Hardware. In einem aktuellen Kernel sollten alle benötigten Treiber bereits als Module mitübersetzt sein. Linux bietet sowohl für einige Wireless PCI-Karten als auch für sehr viele PCMCIA-Karten die notwendige Treiberunterstützung. Wireless PCI-Karten können verwendet werden, um einen Desktop-PC drahtlos mit anderen Rechner zu vernetzen, während →PCMCIA-Karten speziell für Laptops und andere mobile Geräte gedacht sind. Eine weitere Gruppe stellen die USB-basierten Karten dar, deren Bedeutung immer mehr zunimmt. Für Linux gibt es bisher nur für wenige solcher Karten Unterstützung, was sich aber sicherlich im Laufe der Zeit ändern wird.

Die Inbetriebnahme einer Wireless-Karte unter Linux gestaltet sich grundsätzlich sehr einfach und besteht nur aus wenigen Schritten. Zum einen muß der entsprechende Kernel-Treiber für die jeweilige Karte geladen werden. Speziell für PCMCIA-Karten wird diese Aufgabe i. d. R. schon durch die Konfiguration des Linux-PCMCIA-Systems übernommen. In den meisten Fällen muß hier lediglich die PCMCIA-Karte eingesteckt werden, und der korrekte Treiber wird geladen. Bei PCI- und USB-basierten Karten sollte zuerst versucht werden, die Karte mit dem Konfigurationswerkzeug der Distribution (bei SuSE: YaST) zu konfigurieren. Gelingt dies nicht, muß die Auswahl des Treibers manuell, z. B. aufgrund des Kartentyps und Herstellers, erfolgen. Nötigenfalls können einfach verschiedene Treiber ausprobiert werden. An Treibern stehen grundsätzlich drei verschiedene Quellen zur Verfügung: Die Treiber des PCMCIA-Systems, die Kernel-basierten PCMCIA-Treiber sowie Treiber des WLAN-NG-Projekts. Weitere Informationen hierzu finden sich in Abschnitt 7.3.3.2 auf Seite 655.

Neben dem Laden der Treiber sind verschiedene Einstellungen für ein WLAN möglich, die für das Funktionieren sehr wichtig sind. Diese Einstellungen können unter Linux mit Hilfe des Programms `iwconfig` vorgenommen werden, das Teil der sogenannten Wireless Tools ist. Die Wireless Tools sind eine Sammlung von Linux-Programmen und Bibliotheken, die bei SuSE Linux in dem Paket `wireless-tools` enthalten sind. Das Wesentliche an den Wireless-Tools ist die

Tatsache, daß die möglichen Einstellungen mit *einem* Werkzeug für alle Kartentypen gemacht werden. Hierzu wurden im Kernel die sogenannten Wireless Extensions implementiert, letztlich eine Programmierschnittstelle (API), die es dem Benutzer erlaubt, unterschiedliche Treiber für Wireless-Hardware in einheitlicher Weise, z. B. mit `iwconfig`, zu konfigurieren. Die Wireless Extensions wurden als Erweiterung der Netzwerkschnittstellen implementiert, daher stammt auch die Namensanalogie des Werkzeugs zur Konfiguration von Netzwerkkarten `ifconfig` und dem Werkzeug zur Konfiguration von Wireless-Netzwerkkarten `iwconfig`.

Die wichtigsten Parameter, die eingestellt werden müssen, um ein neues Gerät in ein Wireless LAN einzubinden, sind die sogenannte \rightarrow SSID und Einstellungen zur verschlüsselten Datenübertragung mit Hilfe von \rightarrow WEP. Die SSID (Service Set Identifier), manchmal auch als Netzwerk-Name bezeichnet, dient der logischen Trennung von verschiedenen Funknetzwerken. Da Funknetzwerke keine klare räumliche Abgrenzung ermöglichen und sich somit überlappen können, wird eine Zeichenkette als Schlüssel verwendet, um Zugang zu *einem bestimmten* Funknetz zu erhalten. Alle Geräte eines Funknetzes teilen sich diesen gemeinsamen Schlüssel. Die Konfiguration des Access-Points (die Sende/Empfangs-Station, über die eine Anbindung an ein Festnetz durchgeführt wird, s. u.) entscheidet darüber, welche SSID von den mobilen Client-Rechnern verwendet werden muß². Dabei kann ein Access-Point so konfiguriert werden, daß er die SSID ständig mit aussendet, so daß jeder Client sie hören kann. In diesem Fall darf auf dem Client einfach die Zeichenkette `any` als SSID verwendet werden. Strahlt der Access-Point die SSID jedoch nicht aus, so kann die verwendete SSID nur vom Administrator des Access-Points erfahren werden und muß beim Client entsprechend konfiguriert werden. In diesem Fall wird die \rightarrow SSID also als ein erstes Mittel dazu verwendet, den Zugriff auf ein Funknetz einzuschränken, so daß nicht jeder, der mit einem Laptop in der Hand an dem vernetzten Bereich vorbeikommt, das Netz automatisch sieht und direkten Zugriff darauf erhält.

Der zweite für die Inbetriebnahme wichtige Parameter betrifft die Sicherheitseinstellungen für die Funkübertragung. In der Default-Konfiguration werden alle Daten unverschlüsselt übertragen, d. h. jeder, der sich im Bereich des Funknetzes aufhält, kann alle Daten problemlos abhören. Ein Verfahren zur verschlüsselten Datenübertragung, das von allen Geräten unterstützt wird, ist WEP, eine Verschlüsselungsverfahren, das als minimaler Schutz vor Abhörversuchen Dritter gewertet werden kann. WEP basiert auf einem gemeinsamen geheimen Schlüssel, der von allen Geräten geteilt wird. Wer seinen Rechner in ein bestehendes WLAN integriert, muß sich erkundigen, ob WEP verwendet wird. Darüber hinaus muß er noch den WEP-Schlüssel kennen (sozusagen das Passwort). Wer sein

²Wie später noch dargestellt wird, kann ein Funknetz auch ganz ohne Access-Point im sogenannten Ad-Hoc-Modus betrieben werden. In diesem Fall sind lediglich die SSID-Einstellungen der an dem Netz beteiligten WLAN-Karten von Interesse.

eigenes WLAN aufbaut, sollte zunächst für erste Tests ganz auf die Verschlüsselung verzichten und diese erst dann aktivieren, wenn das WLAN grundsätzlich funktioniert. Auch WEP bietet leider keine 100% Sicherheit. Mehr zu diesem Thema wird in Abschnitt 7.3.4.3 auf Seite 678 gesagt.

7.3.2 Struktur eines WLANs

Bevor ganz konkret auf die Konfiguration eines WLANs eingegangen wird, soll zunächst dargestellt werden, welche Varianten es beim Aufbau von WLAN Netzwerken gibt. Ein 802.11 Wireless LAN kann grundsätzlich in zwei verschiedenen Modi aufgebaut bzw. betrieben werden, die als Ad-hoc- und Infrastructure-Mode bezeichnet werden.

Ad-hoc Im Ad-hoc-Modus besteht ein Wireless Netzwerk einfach aus den beteiligten Client-Rechnern, die mit Hilfe einer WLAN-Karte direkt miteinander kommunizieren können. Diese Art der Vernetzung ist in Abbildung 7.4 dargestellt. Zur Vernetzung wird in diesem Fall außer den beteiligten Rechnern und den WLAN-Karten für diese Rechner keine weitere Hardware, wie z. B. eine zentrale Komponente, benötigt. Jeder Rechner, der über eine WLAN-Karte verfügt, kann direkt mit anderen Rechnern des gleichen WLANs, die sich in Reichweite befinden, kommunizieren. Der Ad-hoc-Modus wird in einigen Fällen auch als peer-to-peer-Modus oder als IBSS (Independent Basic Service) bezeichnet. Obwohl IBSS die im Standard offiziell eingeführte

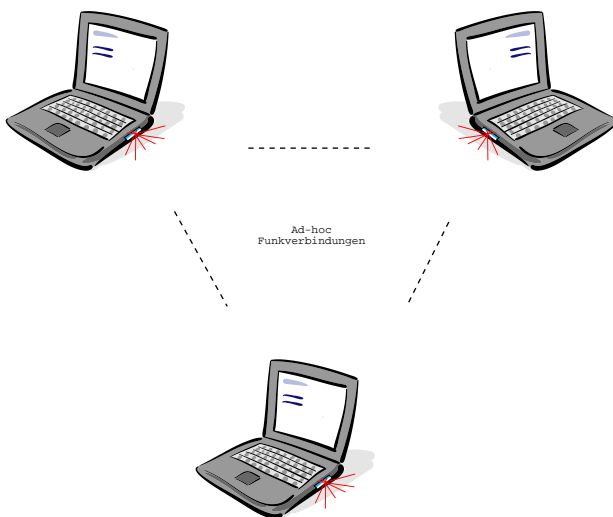


Abbildung 7.4: Der Ad-hoc-Betriebsmodus eines WLANs

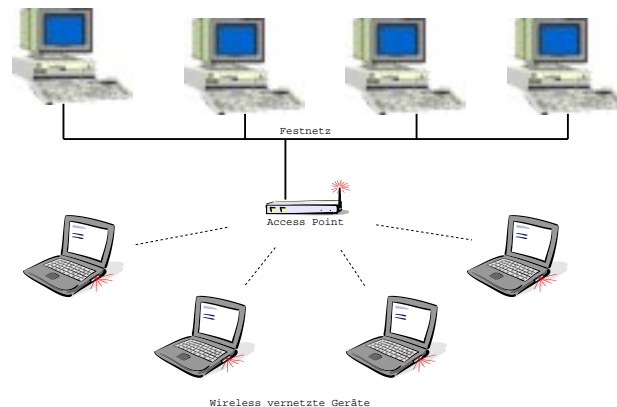


Abbildung 7.5: Der Infrastructure-Betriebsmodus eines WLANs

Bezeichnung für diesen Modus darstellt, wird sie relativ selten verwendet. Daher wird im folgenden nur vom Ad-hoc-Modus gesprochen.

Infrastructure Ein Wireless LAN, das im Infrastructure-Modus (offiziell auch als BSS oder ESS³ bezeichnet) betrieben wird, verfügt im Gegensatz zum Ad-hoc-WLAN über eine oder mehrere, zentrale Komponenten, über die aller Funkverkehr abgewickelt und der Zugriff auf das Medium unter den Anwendern geregelt wird. Eine solche Komponente wird *Access Point* genannt. Ein Access Point ist eine eigenständige Sende/Empfangseinheit, die zusätzlich über eine Anbindung an das Festnetz verfügt. Rechner, wie z. B. Laptops mit WLAN-Karten, die Daten untereinander austauschen wollen, müssen dies im Infrastructure-Modus *immer* über den Access-Point tun, sie kommunizieren also *nicht* direkt untereinander. Eine solche Konfiguration ist in Abbildung 7.5 dargestellt. Neben der reinen Sende- /Empfangsfunktion sowie der Anbindung von Wireless vernetzten Rechnern an ein Festnetz bieten Access Points den Client-Systemen auch die Möglichkeit zu Roamen, was insbesondere für größere Organisationen wichtig ist. Soll z. B. ein ganzer Campus einer Universität drahtlos vernetzt sein, so ist die Ausdehnung des Gebiets, das drahtlos vernetzt werden muß, i. d. R. größer als die Reichweite, die mit zwei Funknetzwerkarten überwunden werden kann. In diesem Fall können einfach ein oder mehrere Access Points auf dem Gelände aufgestellt werden, so daß überall ein ausreichend guter Funkkontakt von einem Access-Point zu einer WLAN-Karte, z. B. in einem Laptop, möglich ist. Ein Client-Rechner, z. B. der eines Studenten, der mit seinem Laptop über den Campus geht, muß daher je nach seinem Standort von unterschiedlichen Ac-

³Der Begriff BSS (Basic Service Set) bezeichnet einen Access-Point mit Anbindung an ein Festnetz, der Wireless Clients bedient. Der Begriff ESS (Extended Service Set) meint ein oder mehrere BSS, die ein gemeinsames Subnetz bilden.

cess Points „bedient“ werden. Mit Roaming wird die Möglichkeit, z. B. eines Laptops, beschrieben im Fall einer Standortveränderung unterbrechungsfrei die Verbindung von einem Access Point an einen anderen zu übergeben. Für den Benutzer des Laptops ist dieses Umschalten transparent und ermöglicht auf diese Weise eine fast beliebige Ausdehnung eines Funknetzes, durch das sich der Anwender frei und ohne Unterbrechung der Netzverbindung bewegen kann. Das Roaming selbst erfolgt automatisch durch die beteiligten Access Points ohne zutun des Anwenders.

Neben den genannten Funktionen statten Hersteller Access Points häufig zusätzlich mit speziellen Funktionen aus, z. B. als Router für einen DSL-Anschluß als auch mit zusätzlicher Firewall-Funktionalität. Als Hardware für Wireless-Funktionalität kommt übrigens im Inneren des Access Points nicht selten eine Wireless PCMCIA-Karte des gleichen Herstellers zur Verwendung.

Aus dem Gesagten ergeben sich ganz klar die Anwendungsszenarien für beide Modi. Sollen nur zwei oder drei Rechner miteinander vernetzt werden, kann die preiswertere Variante der Ad-hoc Vernetzung verwendet werden. Wer zusätzlich eine Anbindung an ein Festnetz benötigt und auch Roaming-Funktionalität benötigt, sollte in einen Access Point investieren und sein Funknetz im Infrastructure Modus betreiben.

Im folgenden wird, falls nichts anderes gesagt wird, davon ausgegangen, daß ein Funknetz im Infrastructure-Modus mit einem Access Point betrieben wird. Daher wird in den folgenden Abschnitten zunächst auf die Konfiguration des Access Points eingegangen und erst dann auf die Konfiguration der Client-WLAN-Karten. Hinweise für die Konfiguration, um ein WLAN im Ad-hoc-Modus zu betreiben, werden in Abschnitt 7.3.3.4 auf Seite 672 gegeben.

7.3.2.1 Hinweise zur Standortwahl eines Access Points

Wer ein Funk-LAN neu aufbaut und dabei einen Access Point verwendet, sollte sich ein paar Gedanken über die Platzierung dieses Geräts machen. Da die Vernetzung über Funkwellen im 2 bzw. 5 GHz-Bereich erfolgt, kann der Standort entscheidend darüber sein, welcher Bereich mit einem Access Point abgedeckt werden kann. Funkwellen in diesem Frequenzbereich werden durch alle festen Substanzen abgeschwächt. Insbesondere Metalle oder metallische Beschichtungen, z. B. auf Folien zur Wärmeisolierungen etc., können die Funksignale fast ganz abschirmen.

Ein Access Point sollte nun einfach so positioniert werden, daß sich möglichst wenig der oben genannten Hindernisse auf dem Weg zu den Client-Rechnern befinden. Auf der anderen Seite muß auch das Festnetz oder der DSL-Anschluß noch über ein Kabel erreichbar sein. Da der Access-Point einen Stromanschluß benö-

Index

Symbole

- .NewsClipper 275
- .cshrc 242
- .faxnrs 189
- .fetchmailrc 327, 328,
331
- .forwad 341
- .hylarc 136
- .lynxrc 244
- .netrc 334
- .ppprc 363
- .profile 242
- .susephone 164
- .vboxrc 483
- .xawtv 518
- /dev/capi20 .. 198, 201
- /etc/XF86Config . 527
- /etc/aliases 139
- /etc/cron.daily . 321
- /etc/crontab 135, 321
- /etc/csh.cshrc .. 242
- /etc/cshrc 319
- /etc/hosts ... 389, 626
- /etc/hosts.deny . 319
- /etc/inetd.conf 122,
399, 400, 402, 471, 484,
485
- /etc/init.d/boot.
local 663
- /etc/inittab 122, 178,
181, 375, 381, 382, 456,
463, 465, 487
- /etc/lilo.conf .. 215
- /etc/modules.conf ..
514, 623, 635
- /etc/nntpsrvr . 319
- /etc/pcmcia/
wireless 659
- /etc/pcmcia/
wireless.opts ...
664
- /etc/ppp/
chap-secrets . 80,
369, 377
- /etc/ppp/ip-down 99,
308
- /etc/ppp/ip-up ... 99,
308, 324, 329, 337
- /etc/ppp/options 20,
29, 47, 73, 363, 364,
376
- /etc/ppp/
pap-secrets ... 80,
365, 366, 368
- /etc/ppp/peers/ . 378
- /etc/profile 242, 319
- /etc/rc.config .. 126,
319, 671
- /etc/rc.d/init.d 137
- /etc/resolv.conf 31,
81, 312
- /etc/services ... 399,
484
- /etc/smpppd.conf 107
- /etc/sudoers 324
- /etc/sysconfig/
network/wireless
siehe WLAN
- /etc/sysconfig/
pcmcia 659
- /proc/interrupts . 60
- DF-Flag 102
- adsl-setup 90
- adsl-start 93
- adsl-status 93
- adsl-stop 93
- cinternet 77
- gnomemeeting
ILS-Browser 604
- ip_dynaddr .. 78, 86, 95
- kinternet 77
- masqmail
Mail-Routing 341
- pppd 74
- smpppd 77
- sudo 68, 84, 94
- /var/log/ppp-usage .
101
- ip-down.local 99
- ip-up.local 99
- ~/ .ppprc 363
- 4:1:1 *siehe* YUV
- 4:2:2 *siehe* YUV
- 802.11 *siehe* Wireless LAN

A

- A-Law-Codierung ... 417

Index

- Access Point *siehe*
Wireless LAN
- Adaptive Answer
Support 149
- Adaptive Answer
Support 695
- ADPCM-Codierung . 417
- ADSL 68, 695
- AIFF 585
- aktion 521, 585
- Analog/Digital-Wandler
416
- Anlagenanschluß 206
- Anrufbeantworter .. 415,
454
- vbox 470
- mit ISDN 470
- vgetty 455
- API 695
- apop 358
- Artefakte 563
- AT-Kommandosatz ... 16
- Audio-CD 434
- audiocd-Plugin 437
- Audiodaten
Dateiformate 418
- Kompression 417
- Konvertierung 421, 425
- AuthAck 66
- AutoPPP 375
- autovbox 427, 485
- AVI 563
- avmcapictrl 53
- B**
- B-Y 566
- bladeenc 436
- BTTV 501
- C**
- c2faxrecv
HylaFAX-Modus .. 195
- Aktivierung 197
- Eingehendes Fax .. 204
- Faxempfang 204
- Parameter für
FaxRcvdCmd ... 205
- Standalone-Modus 195
- c2faxrecv
Aktivierung 201
- c2faxsend
Faxversand 201
- Cache, von wwwoffle ...
292
- call
,Option von
capiplugin ... 378
- callerid.conf 44
- camstream *siehe* Kamera
- CAPI 48, 192, 695
- /etc/ppp/peers . 62
- Übersetzungsvorgang 55
- /etc/capi.conf . 59
- capiinit 61
- capiplugin 67
- download 50
- Faxen 49
- I4L-Kompatibilität .. 61
- Kernel-Modul 54
- Kernel-Optionen ... 56
- Konfigurationsdatei 66
- Konfigurationsdatei für
Einwahl 62
- Module 53
- Optionen 67
- Point-to-Point 61
- pppd-Start als Benutzer
68
- Protokolle 49
- Provider-Konfiguration
62
- selbst übersetzen ... 53
- Treiberinstallation .. 51
- und DDI 60
- und i4l_hardware ..
52
- und rc.config ... 52
- Utilities 54
- capi.conf 207
- capi4hylafax
download 193
- capi4hylafax
- HylaFAX-Spoolverzeichnis
195
- am Anlagenanschluß ..
207
- Automatische
Faxverteilung ... 209
- Einführung 192
- Installation 194
- Installationskript . 195
- Konfiguration,
Anlagenanschluß 207
- Konfigurationsparameter
196
- selbst übersetzen .. 198
- capifax 218
- capifaxrcvd 219
- capiinf 217
- capiinit 53
- capiplugin 377
- cardmgr . *siehe* PCMCIA
- CCITT 113, 696
- cdbakeoven 452
- cdda2wav 435
- cdparanoia 435
- cdrdao 436
- cdrecord 436
- cfaxrecv
Faxempfang 201
- CHAP 696
- chap-secrets 379
- chat 360
- cinternet .. 26, 45, 106,
107
- CLI-Authentifizierung ...
379
- Codec 562
- ConfAck 65
- config.faxCAPI .. 196,
200, 203
- ConfRej 65
- ConfReq 65
- CSI 696
- CSS 581, 696
- CTI 697
- CTS 11

- D**
- Dateiübertragung ... 149
 - Datenübertragung
 - asynchrone 7
 - synchrone 7
 - Datenflußkontrolle
 - Hardware 11
 - Software 11
 - Datenkommunikation ... 111
 - Datenkompression ... 15
 - DB25 8
 - DB9 8
 - DCE 6
 - DDI 697
 - Dial on demand *siehe* DSL
 - Digitalisierung von
 - Audiosignalen .. 416
 - distribfax 210
 - DivX 565
 - DIVX5 576
 - dll.conf 225
 - DMT *siehe* DSL, 697
 - DNS 697
 - dnsmask 314
 - dpi 697
 - Drei-Draht-Leitung 9
 - Drucker, für Fax-Versand
 - 220
 - DSL 697
 - debug-Option 84
 - demand-Option 85
 - idle Parameter 81
 - idle-Option 87
 - modules.conf 80
 - nodetach-Option . 84, 87
 - und IP-Masquerading . 103
 - und Sprachdienste .. 70
 - ADSL 69
 - asymmetrisch 69
 - dial on demand 85
 - Dial on demand 78
 - DMT 70
 - Ethernetkarte 74
 - Frequenzspektrum . 70
 - HDSL 69
 - kernelbasiert 74
 - Kostenkontrolle 99
 - manuelle Konfiguration
 - 80
 - mit yast2 75
 - MTU-Problem 101
 - Nameserver 81
 - Nameservers 85
 - RP-PPPoE 88
 - tkppoe 97
 - dial on demand 94
 - graphische
 - Konfiguration 97
 - script-Konfiguration . 90
 - SDSL 69
 - Splitter 89
 - symmetrisch 69
 - T-DSL *siehe* T-DSL
 - U-R2 Standard 72
 - Varianten 69
 - VDSL 69
 - DTE 6
 - DTMF 698
 - DVD 564, 565
 - dvips 176
 - dvr 567

E

 - ECP 622, 698
 - EIA 117, 698
 - Einwahl 104
 - über T-DSL 73
 - mit Modem 20
 - Email 325
 - vom ISP holen 326
 - EPP 622, 698
 - EPS 698

F

 - Fax
 - Grundlagen 112
 - Gruppe 3 113
 - Gruppe 4 113
 - Klasse 1 127
 - Klasse 1...2.0 115
 - Komprimierung ... 114
 - mit Modem 111
 - on Demand 118
 - Phasen A...E 114
 - polling 118
 - Standards 113
 - Fax-Drucker 220
 - Fax-Klasse 1 698
 - Fax-Klasse 2 und 2.0 . 699
 - faxaliases 189
 - FAXCOVER 175
 - Faxdienst 206
 - faxgetty 197
 - faxheader 189
 - faxnotify.capi .. 210
 - faxprint 222
 - faxprt 223
 - faxq 187
 - FaxRcvdCmd 205
 - faxrm 187
 - faxrunq 187
 - faxspool 187
 - Optionen 189
 - FBAS 503
 - Fehlerkorrektur 15
 - Fernsehen 501
 - Grundlagen 515
 - kwintv 523
 - xawtv 517
 - Vollbildmodus 517
 - fetchmail 356
 - .fetchmailrc ... 327
 - Passworte 328
 - automatisch Mail holen
 - 329
 - Benutzer-Optionen 334
 - Einführung 326
 - globale Optionen .. 331
 - Konfiguration 327
 - multidrop-Mode .. 338
 - plugin 336
 - Protokolle 326
 - Server-Optionen ... 332
 - sichere Übertragung ... 335

- und mehrere Provider .
337
- ssh 336
- fetchmailconf 329
- fetchnews 318, 320
- ffmpeg 561, 569
- ffplay 569
- ffserver 569
- FIR 629, 699
- Firmware 699
- Firmware-Datei 207
- Foren *siehe* News-Gruppe
- Frame 563
- FreeS/WAN 685
- FTP 247
 - Server 397
 - anonymous-Zugang ..
397
 - Download 247
 - dynamische
Server-Merkmale 253
 - Port 248
 - Upload 247
- ftpd 398
- G**
- G.711 *siehe*
Video-Konferenz
- G.723.1 *siehe*
Video-Konferenz
- G.728 *siehe*
Video-Konferenz
- G.729 *siehe*
Video-Konferenz
- G3-Fax-Format . 180, 184,
185, 187, 190
- Gatekeeper *siehe*
Videokonferenz
- Gateway *siehe*
Video-Konferenz
- GIMP 224
- gmplayer 577
- gnomemeeting 599
 - H.245-Tunneling .. 607
 - Text-Chats 605
 - und Firewalls 608
- Verbindungsaufbau ...
602
- grip 439
- Grundkonfiguration ... 5
- Gruppe 4 *siehe*
Fax-Gruppe 4
- Gruppe 3 *siehe*
Fax-Gruppe 3
- Gruppe 3 Fax 699
- Gruppe 4 Fax 699
- gs
 - Fax-Konvertierung 202
- GSM 06.10 *siehe*
Video-Konferenz
- GSM-Standard 418
- H**
- H.225 *siehe*
Video-Konferenz
- H.235 *siehe*
Video-Konferenz
- H.245 *siehe*
Video-Konferenz
- H.261 *siehe*
Video-Konferenz
- H.323 *siehe*
Video-Konferenz
- Handshake *siehe*
Datenflußkontrolle
- Hayes 16
- HDLC 360, 378, 699
- hfaxlclient 223
- hfaxlif 223
- hfaxlserver 223
- hisax.o 40
- HylaFAX 119
 - /etc/inittab ... 137
 - Datenflußkontrolle 127
 - Datenverbindungen ...
149
 - Deckblätter 173
 - Dialrules 142
 - dialtest 144
 - eigenes Deckblatt .. 174
 - eingehendes Fax .. 139
 - /etc/hosts.hfaxd
125
- ./etc/hosts.hfaxd
147
- FAQ 141
- Fax-Server start/stop ..
137
- faxaddmodem 124
- faxalter 156
- Faxauflösung 120
- faxcover 173
- FAXCOVER-Environment-
Variable 173
- faxcron 135
- faxd 122
- faxgetty 122, 137
- Faxmaster 139
- faxmodem 127
- faxqclean 135
- faxrm 139, 156
- FAXSERVER-Variable ..
156
- faxsetup 127
- faxstat 139, 156
- Fehlersuche 140
- Funktionsdiagramm ...
121
- hfaxd 121
- Installation 123
- Konfiguration,
Übersicht 123
- Konfigurationsdateien .
135
- Mailing-Liste 141
- Modem-Einrichtung ...
132
- Modeminstallation 125
- Modemkommunikation
122
- Ports 122
- Probleme mit
masqmail 154
- sendfax-Optionen ...
153
- sendfax 139, 152
- Serverkonfiguration ...
127

- Serverzugriffskontrolle
 147
 TSI-basierte
 Faxunterdrückung ..
 146
 TSI-basierte Zustellung
 145
 Verzeichnisstruktur 124
 Web-Seite 119
 Zielnummernkontrolle
 148
 HylaFAX
 hyla.conf 136
- I**
- I4L 700
 I4L-Projekt 40
 ICMP 102
 IETF 700
 ifup-Skript 659
 ILS 599
 IMAP 700
 imon 48
 imontty 48
 info 181
 inittab
 mgetty 181
 insserv 356
 Intercast-Programme 501
 Intercasting 544
 Internet
 Fax versenden 227
 ip-down 389
 IP-Telefonie *siehe*
 Video-Konferenz
 ip-up 349, 389
 ip-up.local 349
 IPCP 66, 361, 700
 ippd 360
 IPSEC 685
 irattach 632, 637
 IrDA 628, 700
 Übertragungsgeschwindigkeit ... 628
 Betrieb im FIR-Modus .
 636
- Betrieb im SIR-Modus .
 635
 Betriebsmodi 629
 Dongle-Betrieb 636
 Einführung 628
 Interfaces 639
 Kernelkonfiguration ...
 632
 Konfiguration 631
 Protokolle 629
 serielle
 Kommunikation 640
 Utilities 633
 irda-utils 633
 irlan 637
- ISDN**
- Karte 39
 aktive Karte 192
 aktive Karte 39
 Anlagenanschluß ... 38
 B1-Basiskonfiguration .
 216
 B1-Firmware 217
 B1-Kernel-Konfigu-
 ration 213
 Basisanschluß 37
 CAPI 49
 D-Kanal 38
 Einführung 36
 Einwahl 46
 Fax-Versand über .. 191
 Kartenkonfiguration 40
 Kernel-Dokumentation
 192
 Kernel-Konfiguration ..
 41
 Konfiguration mit
 yast2 42
 Mehrgeräteanschluß 38
 MSN 38
 passive Karte 39
 passive Karte 192
 Primärmultiplex-
 Anschluß 39
 Utilities 48
 isdn.conf 44
- isdnctrl 48
 ISP 700
 ITU 113, 700
 iwconfig *siehe* Wireless
 LAN, *siehe* Wireless
 LAN, 672, 680
 iwlist 687
 iwpriv 688
- J**
- Job-Nummer 153
 JVTxView 546
- K**
- kam2 491
Kamera
 an TV-Karte 551
 an USB 551, 553
 camstream-Anwendung
 556
 Merkmale 550
 Philips ToUcam ... 553
 Treiber-Parameter . 554
 und Linux 550
 USB-Treiber 553
 Web-Cams 552
- KDE**
- Faxdruck 220
 kdeprintfax 221
 keafnode 322
 kermit 238
Kernel-Konfiguration
 fürB1-ISDN-Karte . 213
 kfax 171, 184
 kfm 247
 khylafax 169
 kimon 48
 kinternet .. 26, 45, 106,
 107
 koncd 451
Konfiguration
 des Proxy in Netscape .
 291
 konqueror 437
Konvertierung
 des G3-Fax-Format 185
 kppp

- Einführung 29
 Einwahl 35
 Konfiguration 30
 krecord 433
 ksendfax 166, 190
 kvoice 455
 kwifimanger 686
 kwintv 524
 Kanaleditor 526
 Quick-Access-Leiste ...
 525
 Snapshots 527
 Videoaufzeichnung 527
 kwvdial 24
- L**
- lame 435, 569
 L^AT_EX 176
 latex-cover
 Fax-Deckblatt erstellen
 mit 175
 latex-cover.sty . 176
 LCP 65, 361, 701
 leafnode
 applyfilter 325
 Arbeitsweise 317
 automatisch Artikel
 holen 324
 checkgroups 325
 Einführung 317
 Initialisierung 320
 Installation 318
 Konfiguration 318, 319
 Konfigurationsvariablen
 322
 newsq 325
 texpire 320
 und inetd.conf . 318
 libtiff 123
 LIRC
 Installation 528
 LIRC-Projekt 528
 lirc_i2c 533
 lircd 532
 lircd.conf 532
 lpi 701
 lynx
- Bookmarks 245
 Einführung 241
 History 245
 Konfiguration 242
 Navigation 244
 Tastaturbelegung .. 246
 wwwoffle 242
 und Proxies 243
- M**
- masqdiabler 105
 masqmail 340
 example.route . 350
 Adress-Rewriting . 351
 Arbeitsweise 341
 externe, interne Mail ..
 341
 Features 340
 Grundkonfiguration ...
 342
 Installation 342
 Log-Meldungen ... 356
 Mail abrufen 356
 Mail ausliefern 356
 Mail-Routing . 348, 349
 Online-Zustand ... 348
 Start-Skript 355
 und procmail 341
 und HylaFAX 154
 Verwendung von .. 354
 masqmail
 masqmail.conf . 343
 MCU *siehe*
 Video-Konferenz,
 siehe Video-Konferenz
 seeVideo-Konferenz . 589
 Mehrgeräteanschluß . 206
 mgetty 177
 login.conflogin.conf
 376
 Fax-Empfang 184
 Konfigurationspara-
 meter 180
 mgetty+sendfax
 graphische Frontends .
 191
 mgetty-sendfax
- Grundkonfiguration ...
 178
 mgetty.conf 178
 MH 114
 minicom 13, 126, 231
 Bedienung 234
 Dateitransfer 238
 Grundkonfiguration ...
 231
 Tastaturbedienung 235
 Verbindungsaufbau ...
 236
 Mirror 261
 Mitbenutzernummer . 80
 mkisofs 436
 modem, symbolischer
 Link 178
 Modem 12, 701
 Datenkompression . 15
 Fehlerkorrektur 15
 Grundlagen 5
 Modulationsverfahren .
 14
 V-Standards 14
 Modulationsverfahren 14
 Motion JPEG 567
 motv 521
 MP2 585
 MP3 418, 581, 585
 MPEG 564
 MPEG-1 ... 563, 564, 576,
 581
 MPEG-2 564, 576
 MPEG-4 565
 MPEG-Standard 418
 mpg123 431
 mplayer 576
 MR 114
 MSN 379, 701
 MTU *siehe* DSL, 103
 Multimedia-Karte ... 503
 Geräte-dateien 506
 Kernel-Konfiguration ..
 506
 Konfiguration 505
 Modulmeldungen . 513

- Treiber-Konfiguration .
507
modules.conf ... 514
- N**
- Namensserver 312
ncftp 248, 249, 256
Arbeitsverzeichnis 250
Benutzeroberfläche 250
Datei-Download .. 252
FTP-URL 256
History-Funktion .. 251
Kommandoliste ... 255
Kommandos 251
Konfiguration 254
Modus 249
Optionen 257
Statuszeile 250
Upload 253
Wildcards 252
NCP 66, 361
Net-Meeting
und Fast-Connect . 607
und Firewall 609
netpbm 185
netscape
Proxy-Konfiguration ..
291
new_fax,mgetty-Skript
185
News-Gruppe 316
News-Reader 316
NewsClipper
Beispiel 269
Einführung 268
Handler 272, 279
Handler erstellen .. 280
Konfiguration 273
Konfigurationsdatei ...
275
MakeHandler.pl 284
Start/Stop-Muster . 283
Newsticker.pl 271
NIS 701
NNTP 317, 701
- NNTPSERVER,
Environment-Variable
319
noatun 432, 581
NTSC 516, 702
Nullmodem-Kabel .. 361,
362
Nullmodem-Kabel ... 10,
126, 230, 238, 395
- O**
- oggenc 436
OggVorbis 581
ogle 584
ohphone 609
openmcu *siehe*
Video-Konferenz
- P**
- P2P *siehe*
Anlagenanschluß
PAL 516, 702
PAP 702
PAP-Protokoll 365
pap-secrets 379
Parallel-Port 624
Paritäts-Bit 7
PCI-WLAN-Karte ... 655
PCMCIA 702
/etc/pcmcia/
network.opts 664,
671
/etc/sysconfig/
network 672
/etc/sysconfig/
network/wireless
659, 666
cardctl 665
cardctl ident . 690
cardmgr . 657, 665, 689
config-Datei 690
Konfiguration 689
network-Skript .. 658,
665
network.opts ... 658
Problembehandlung ...
689
- stab-Datei 690
wireless-Skript . 665
WLAN-Karte . 654, 656
play 429
PLIP 702
Einführung 622
Interface-Konfiguration
625
Interrupt 624
Kabel 622, 627
Kernel-Konfiguration ..
622
Konfiguration 622
Port-Basisadresse .. 624
Route anlegen 626
pnmtops 225, 227
pop 358
POP 703
PPP 19, 360, 702
Kernelsupport 360
Netzwerk-Interface 361
Skripte 387
Authentifikation .. 368
CHAP-Authentifikation
371
chap-secrets .. 366,
372
chat-Skript 366
Client-Konfiguration ..
364
dynamische Adresse ..
362
Einführung 360
Einsatzgebiete 362
IP-Adressangabe .. 377
ip-up 388
LAN-Kopplung ... 393
ttyI-devices 382
Netzwerk-Interface 368
Optionen 384
Optionsdateien 363
PAP-Authentifikation .
369
pap-secrets ... 369
pap-secrets 366

- privilegierte Optionen .
363
- proxyarp 393
- Rechner-Rechner Verb.
361
- Rechner-Rechner
Verbindung 395
- Routing 394
- Server-Konfiguration ..
373, 377
- main 376
- synchrones 46, 382
- und CAPI 377
- mgetty 375
- pppd 360
- PPPoE 73, 101, 102
- proftpd 398
- Proxies
des ISP 291
- Proxy
des ISP 293, 299
- wwwoffle 290
- Proxy-Skript 305
- proxy.pac 303
- PVF-Tools 426
- pvftornd 462
- pwc 554
- pwcx-i386 554
- Q**
- qmail 340
- qtvision 524
- QuickCam 552
- R**
- R-Y 566
- raccess4vbox 492
- Radio 501
- fmttools 540
- kderadio 542
- kradio 541
- RAS *siehe*
Video-Konferenz
- rc.dialout 108
- Real Media 564
- realplayer 564
- rec 429
- Rechner-Rechner
Kommunikation 395
- RFC 703
- RGB 565
- ripit 441
- Rippen 437
- Roaming 648
- rp-pppoe 89
- RP-PPPoE *siehe* DSL
- rsynth 467, 468
- RTP 596
- RTS 11
- RTS/CTS 127, 703
- rz 238
- S**
- Sampling 416
- Sampling-Rate 416
- SANE 224
- Satelliten-Fernsehen . 501
- say 468
- scancopy 225
- scanimage 225
- Scannen, Werkzeuge . 225
- Scanner 224
- scanner
/dev 225
- Schnittstelle
serielle 6, 230
- serielle, Gerätedatei 12
- serielle, Kabel 8
- SECAM 517, 703
- sendfax ... 139, 177, 221
- Fax-Versand .. 183, 185
- Konfigurationspara-
meter 182
- Optionen 186
- sendfax,
mgetty+sendfax ..
187
- sendfax.conf 178
- sendmail 340
- set-tv 522
- SFF, Format 202
- SFF, Bilddateiformat . 218
- SFF-Format 204
- sff2misc 219
- sfftobmp 219
- SIP *siehe*
Video-Konferenz
- SIR 629, 703
- SLIP 704
- smpppd 26, 106
- SMTP 704
- SOHO 704
- sox
Aufrufsyntax 422
- Auto-Modus 425
- Einführung 421
- Optionen 422
- Splitter *siehe* T-DSL
- SSID *siehe* Wireless LAN,
704
- Standards
Fax 113
- stty 13
- sudo 324
- SuSEFax
Einstellungen 158
- Fax versenden 162
- Fax-Polling 164
- Grundkonfiguration ...
157
- Job-Konfiguration . 160
- Serienfaxe 165
- Telefonbuch 164
- SuSEFirewall 684
- SuSEVboxConf 489
- SVCD 565
- SyncPPP 46
- sz 238
- T**
- T-DSL 68, 71
- Übertragungsraten . 73
- options-Datei 80
- Grundlagen 69
- Konfiguration ... 74, 88
- Login-Zeichenkette . 80
- Modem 72
- Splitter 72
- Verkabelung 71
- T-DSL-Modem *siehe*
T-DSL

- T-Online-Anschlußkennung
80
- T.30 704
- T.30 Standard 114
- T.4 704
- T.4 Standard 113
- Terminal *siehe*
Video-Konferenz
- Terminalprogramme . 230
- texpire .. *siehe* leafnode
- tgif
Fax-Deckblatt erstellen
mit 175
- TIA 117, 704
- TIFF, Format 202
- TIFF, Format 204
- TIFF/F-Format . 113, 123,
152, 171, 180, 184
- TSI 145, 705
- U**
- U-Law-Codierung ... 417
- UART 6
- URL 705
- USENET 316, 705
- V**
- V-Standards 113
- V4L 502, 705
- V4L2 502, 706
- VBI 706
- vbidecode 523
Anzeige der Seiten 546
- vbox 470
- vbox 427, 490
.vboxrc 483
- Kernel-Konfiguration ..
471
- Abspielen von
Nachrichten 487
- Benutzer-Dateien .. 478
- Benutzer-Steuerung ...
488
- DTMF-Töne 498
- Erstellen von Ansagen .
485
- Fernabfrage 492
- Installation 471
SuSEVboxConf ... 489
- Konfigurationsschritte .
471
standard.tcl ... 486
kam2 491
vbox 490
vboxd.conf . 476, 479
vboxgetty.conf 472
- Vbox 470
vboxgetty 487
vboxplay 431
vboxtoau 427
- Vernetzung 621
- VFIR 629, 706
- vgetty 177, 455
Ansage erstellen ... 461
Auswertung von
DTMF-Tönen ... 467
Fernabfrage 467
Installation 455
kvoice 465
kvoice 456
- Konfigurationsvariablen
457
- Log-Meldungen ... 466
- Nachrichten abspielen
464
Start von 463
- vgrabbj 559
- Video
Aufzeichnen .. 567, 571
Interframe-Kodierung .
563
Komprimierung ... 562
Streaming 572
- Video for Linux 502
- Video-Konferenz
Einführung 587
G.711-Codec .. 590, 597
G.723.1-Codec 597
G.728-Codec 597
G.729-Codec 597
GSM 06.10-Codec . 597
H.225-Protokoll ... 594
H.235-Protokoll ... 594
- H.245-Protokoll ... 594
H.261-Codec .. 591, 597
H.263-Codec 597
H.323 590
H.323 Gateway 593
H.323 Gatekeeper . 592
H.323-Architektur . 591
H.323-MCU ... 593, 613
H.323-Protokoll ... 589,
594
H.323-Terminal 592
- Megaco-Protokoll . 589,
595
- MGCP-Protokoll .. 595
openmcu 613
- Protokolle 589
- RAS-Protokoll 594
- SIP-Protokoll . 589, 595
- Systeme 588
Voraussetzungen .. 598
- Videokonferenz
H.323-Terminal 590
- Videotext 501, 542
Decoder 543
Gerätedatei 543
vbidecode 544
alevt 543
Seitenkonvertierung ...
523
viewfax 171, 184
- VJ-Compression 361
- vm 462
- VPN 685
- vtx.cgi 546
vtx.sh 546
vtx2ascii 546
- W**
- Warteschlange 156
- WAV 581, 585
wavplay 430
wavrec 430
- WECA 644, 706
- WEP *siehe* Wireless LAN,
706
- wget
.wgetrc 264

- absolute Links 264
- accept-Listen 262
- Authentifizierung . 264
- Domain-
 - Einschränkungen 263
- Einführung 259
- folge FTP-Link 263
- Mirror erstellen ... 262
- Optionen 265
- Reject-Listen 262
- Rekursionsebenen . 261
- rekursiver Download .. 261
- relative Links 264
- Server-Hosts 263
- Timestamps 261
- Verzeichnislisten .. 264
- wgetrc 264
- WIFI 644, 706
- WinFlex 122
- Wireless LAN
 - Übertragungsraten 643
 - ESS 648
 - IBSS 647
 - iwconfig,WLAN-Status 662
 - peer-to-peer-Modus ... 647
 - Standards 643
 - Treiber 655
 - wireless-Skript . 658
- Wireless LAN
 - Managed 662
 - 802.11a 644
 - 802.11g 644
 - Access Point,
 - Plazierung des .. 649
 - essid-Parameter . 661
 - Firewall 683
 - IP-Konfiguration .. 670
 - iwconfig 645, 660
 - Prism-Chipsätze ... 673
 - Reichweite 644
 - restricted-Mode ... 682
 - SSID 646, 675
 - WEP 662
 - WEP-Plus 679
 - wireless-tools 657
 - wlan-ng-Skript ... 674
- Wireless LAN 643
 - 802.11b 643
 - Access Point 648
 - Access Point,
 - Konfiguration ... 650
 - Access Points und DHCP 652
 - Access-Point, externe Antenne 650
 - Access-Points, Filter ... 683
 - Ad-hoc 647
 - Ad-hoc-Netz 672
 - BSS 648
 - commit 662
 - DHCP-konfiguration .. 671
 - Frequenzbereich ... 644
 - Infrastructure 648
 - iwconfig 657
 - Kanäle 644
 - Kanal 654
 - Kompatibilität 644
 - Linux-Treiber 645
 - MAC-basierte
 - Zugangskontrolle ... 676
 - open-Mode 682
 - PCI-Karte,
 - Inbetriebnahme . 655
 - PCMCIA,
 - Inbetriebnahme . 656
 - Sicherheit 674
 - SSID 653, 661, 663
 - SSID im Ad-hoc Netz .. 672
 - Treiber 654
 - und Linux 645
 - WEP 646, 653, 678
 - WEP, im Ad-hoc Netz . 672
 - WEP, Konfiguration ... 680
 - WEP-Schlüssel 678
 - wireless,
 - Einstellungen ... 668
 - wireless-Skript *siehe* PCMCIA
 - wireless.opts,
 - Einstellungen ... 665
 - WLAN-NG-Treiber 673
 - WPA Verschlüsselung . 680
- Wireless Tools 645
- WLAN *siehe* Wireless LAN
- WLAN-NG 655
- World Wide Web 290
- wu.ftpd 398
- WUFTP-Server
 - /etc/ftpaccess 399, 404
 - /etc/ftpusers . 412
 - Benutzerklassen ... 405
 - chroot 400
 - dynamische
 - Konvertierungen 412
 - Funktionsweise ... 398
 - Installation & Konfiguration ... 401
 - wvdial 106
 - Einführung 20
 - Einwahl 25
 - graphische
 - Konfiguration 24
 - Konfigurationsdatei 22
 - Parameter 23
 - wvdial.conf 106
 - wvdialconf 20
 - wwwoffle 294
 - Arbeitsmodi 292
 - Arbeitsmodi
 - automatisch setzen .. 307
 - Arbeitsmodus
 - einstellen 295
 - Autodial-Modus .. 292

-
- Cache-purging 314
DNS-Problem 312
Einführung 290
Funktionsweise ... 291
in einem Mininetz . 310
Installation 296
Konfiguration 296
Konfiguration über
 Browser 296
Konfigurationsvariablen
 300
Loglevel einstellen 299
mehrere Provider .. 308
Monitor 294
Offline-Modus 294
Online-Modus 293
Passwortschutz der
 Konfiguration ... 307
- Proxy-Konfiguration ..
 303
Purge des Cache ... 303
Registrierung zum
 Download 294
Seiten-Optionen ... 294
Statusmeldungen .. 299
wwwoffle.conf 296
wwwoffled
 Proxy 292
- X**
xanim 521, 585
xawtv 518, 554, 556
 Audioaufzeichnungen
 mit record 522
Benutzereinstellungen .
 519
channel Editor 521
- Video erstellen 521
xine 581
xmcid 441
xmms 433
xmovie 585
XON/XOFF 127, 707
xscanimage 226
- Y**
yast2 42
 Modemkonfiguration .
 26
YUV 503, 566
YUV-Kodierung 563, 707
- Z**
Z-Modem-Protokoll . 149,
 238