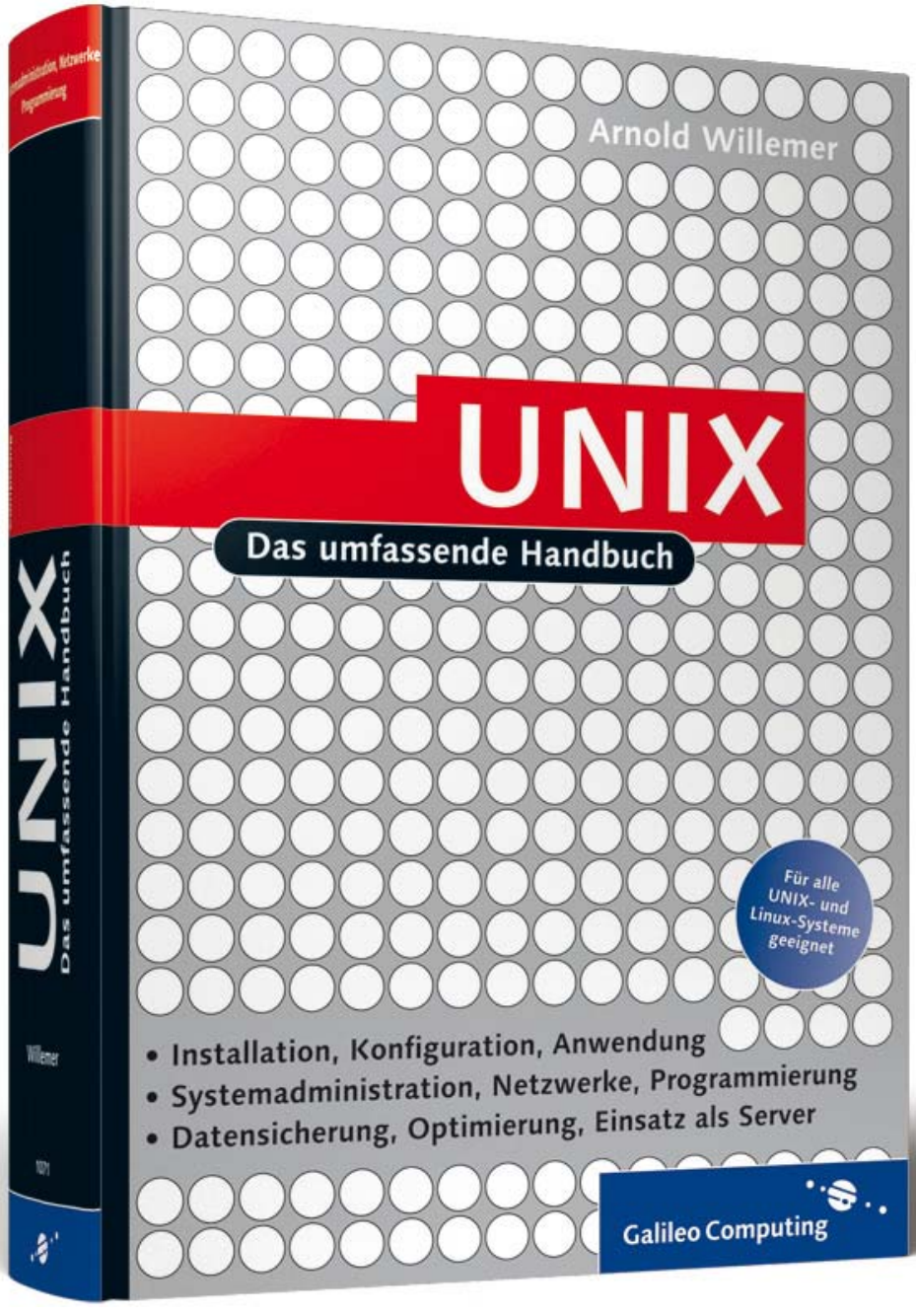
Arnold Willemer
UNIX
Das umfassende Handbuch
Für alle UNIX- und Linux-Systeme geeignet
• Installation, Konfiguration, Anwendung
• Systemadministration, Netzwerke, Programmierung
• Datensicherung, Optimierung, Einsatz als Server
Galileo Computing
UNIX
Das umfassende Handbuch
Willemer

Inhalt

TEIL IV: NETZWERK

TEIL V: UNIX ALS SERVER

TEIL VI: DAS X WINDOW SYSTEM

TEIL VII: PROGRAMMIERUNG

Die Zeit der zentralen Großrechner mit den passiven Terminals klingt aus. Gerade UNIX findet heute seinen Haupteinsatz als leistungsfähiges und zuverlässiges Serversystem in einer Netzwerkumgebung.

18 Netzwerk

Ein Netzwerk ist zunächst nichts anderes als die Verbindung mehrerer Computer, über die sie miteinander kommunizieren können. Das Verbindungsmedium war ursprünglich ein einfaches abgeschirmtes Kabel. Heute besteht ein Netzwerk aus Glasfasern, aktiven Switches oder Funkfrequenzen. In Kombination mit geeigneter Software können Sie von Ihrem Arbeitsplatz aus Ressourcen anderer Computer nutzen. Am bekanntesten dürfte die Möglichkeit sein, gemeinsam auf eine Festplatte oder einen Drucker zugreifen zu können. Das Motiv ist nicht allein Sparsamkeit. Netzwerke ermöglichen das Teilen von Daten und gewährleisten so höchste Aktualität.

Server

Auf den ersten Blick scheint es, als würde ein Netzwerk Objekte wie Drucker oder Festplatten zur Verfügung stellen. Bei näherer Betrachtung handelt es sich aber um *Dienste* (engl. *services*). So werden die Druckdaten nicht an den Netzwerkdrucker selbst gesendet, sondern an einen Prozess. Dieser veranlasst dann als Dienstleister (engl. *server*) den Druck. Dieser Prozess muss nicht zwingend auf einem Computer laufen. Bei den heutigen Netzwerkdrucker läuft nicht nur eine Steuerelektronik, sondern ein kleiner Computer, der die Netzwerkkommunikation beherrscht. Da die Druckdaten nicht direkt in den Druck gehen, kann der Prozess vor dem Druck die Berechtigungen prüfen. Der teure Farblaserdrucker der Werbeabteilung soll beispielsweise nicht allen Angestellten zur Verfügung stehen, um ihre Urlaubsfotos auszudrucken.

Internet

Das größte aller Netze ist das Internet. Hier gibt es Abertausende von Dienstanbietern. Neben dem World Wide Web bietet es eine schnelle, kostengünstige Kommunikation per E-Mail. Darum ist eine Firma in der Regel daran interessiert, vielen Arbeitsplätzen einen Zugang zu dieser Informationsquelle zu ermöglichen. Allerdings besteht auch das Risiko, dass das eigene Netz über das Internet ausgespäht oder angegriffen wird.

TCP/IP

Jedes Netzwerk braucht Protokolle, die festlegen, welcher Teil der Nachricht die Adresse, der Absender, eine Kontrollinformation bzw. Daten ist. Inzwischen ist TCP/IP (Transmission Control Protocol/Internet Protocol) das unangefochten wichtigste Protokoll. TCP/IP ist für UNIX nicht nur das Zugangsprotokoll zum Internet, sondern auch die Basis für lokale Netzwerke. Dabei spielen die vom PC her bekannten Festplatten- und Druckserver keine so große Rolle wie das Verteilen von Anwendungen in Client-Server-Architekturen. In UNIX-Netzen geht es mehr um das Starten von Prozessen auf entfernten Maschinen oder um das Verteilen von Prozessen auf mehrere Maschinen.

Heterogene Netze

In TCP/IP-Umgebungen finden Sie selten ausschließlich UNIX-Maschinen. So haben Sie immer wieder damit zu tun, auch MS-Windows- oder Mac-OS-Rechner zur Zusammenarbeit zu bewegen. Sie werden oft als Frontend benutzt, während das Backend unter UNIX läuft.

18.1 Client-Server-Architekturen

Clients und Server sind Prozesse

Der Zweck eines Netzes ist es, einen Prozess auf einem Rechner mit Informationen zu versorgen, die auf einem anderen Computer vorhanden sind. Der Auslöser ist also immer ein Prozess, der eine Information anfragt und auf die Antwort wartet. Einen »Anfrager« nennt man Client, einen »Antworter« nennt man Server. Ein Server ist also in erster Linie ein Prozess und kein Computer. Derselbe Computer kann durchaus gleichzeitig als Client und als Server auftreten, indem er bestimmte Anfragen beantwortet und auf der anderen Seite Anfragen stellt.

Eine Software, die nach dem Client-Server-Prinzip arbeitet, ist auf zwei Seiten aufgeteilt. Sie ist quasi irgendwo »durchgesägt«. Das Frontend läuft auf dem Arbeitsplatzrechner, und ein anderer Teil arbeitet auf einem anderen, typischerweise zentralen Rechner, der dann auch meist als Server bezeichnet wird, weil auf ihm die Serverprozesse laufen.

18.1.1 Ethernet als Verkabelungsbeispiel

Bei der Verkabelung wird heutzutage meistens Ethernet eingesetzt. Diese Technik ist für lokale Netzwerke inzwischen nahezu konkurrenzlos. In seiner ursprünglichen Form bestand ein Ethernet aus einem Koaxialkabel[1] und je einem Widerstand an jedem Ende. Ein Computer wurde mit dem

1 Ein Koaxialkabel besteht aus einem Draht, der von einer Abschirmung umgeben ist. Antennenkabel sind typischerweise auch Koaxialkabel.

Kabel durch einen Abgriff verbunden. Bei dem Koaxialkabel RG58 war dies ein auf dem BNC-Stecker basierendes T-Stück. Dieses T-Stück steckt direkt auf einem Transceiver am Ethernet-Controller.

Twisted Pair

Inzwischen verwendet man auch in kleinen Netzen längst eine Twisted-Pair-Verkabelung. Das Kabel besteht, wie der Name schon sagt, aus verdrillten Drähten. Es verbindet den Computer mit einem Hub oder einem Switch. der das Rückrat des Netzwerks darstellt. Moderne Netzwerkinterfaces können auch direkt miteinander verbunden werden. Sie schalten ihre Belegung um, wenn sie feststellen, dass sie mit einem anderen Netzwerkinterface statt mit einem Hub verbunden sind.

Jedes Netzwerkinterface hat seine eigene, weltweit eindeutige Nummer, die 48 Bit groß ist. Diese Nummer ist meist in das ROM des Controllers eingebrannt.[2] Der Ethernet-Controller lauscht die ganze Zeit am Kabel, und sobald ein Paket kommt, das die Nummer des Controllers als Adresse hat, holt er es in seinen Speicher und gibt es an das Betriebssystem weiter.

Senden

Will der Controller selbst Daten senden, packt er sie in Pakete zu je maximal 1500 Byte und setzt die Ethernet-Adresse des Zielcomputers, gefolgt von der eigenen Adresse, als Absender hinein. Zu guter Letzt enthält jedes Paket eine Prüfsumme, die CRC (Cyclic Redundancy Check), und schon ist das Paket sendebereit.

Wenn Sie sich im Detail dafür interessieren, was auf Controller-Ebene passiert, wie die Pakete genau aussehen und wie die Prüfsummen arbeiten, sollten Sie die Werke von Tanenbaum[3] und Comer[4] lesen.

18.1.2 Die Pseudoschnittstelle loopback

TCP/IP wird nicht nur verwendet, um mit anderen Rechnern Verbindung aufzunehmen. Manchmal führt der Rechner auch Selbstgespräche. So arbeitet die grafische Oberfläche von UNIX über TCP/IP. Dabei erfolgt die Kommunikation auf modernen Workstations vom eigenen Rechner zum eigenen Display. Bei Maschinen, die überhaupt keine Netzwerkanschlüsse haben, wird eine Schnittstelle gebraucht, über die der Rechner mit sich selbst Kontakt aufnehmen kann. Diese Schnittstelle nennt sich loopback, weil sie wie eine Schleife auf den Rechner selbst zurückführt.

2 Bei Sun-Maschinen befindet sie sich in einem batteriegepufferten RAM. Darum empfielt es sich, sich diese Nummer zu notieren. Die Batterie könnte ja mal schlappmachen. Die Nummer wird beim Booten angezeigt.

3 Andrew S. Tanenbaum: Computer Networks. Prentice Hall, Englewood Cliffs, 1987.

4 Douglas E. Comer: Internetworking with TCP/IP. Prentice Hall, 2nd ed., 1991.

18.1.3 Pakete in Paketen

ARP

Auf der Basis von Ethernet-Paketen werden IP-Pakete versendet. Als Adresse werden IP-Adressen und der Port verwendet (siehe unten). Dabei kennt ein Ethernet-Controller nur MAC-Adresse. Die IP-Adressen müssen also auf die MAC-Adresse abgebildet werden. Das Protokoll, das diese Abbildung steuert, nennt sich ARP (Address Resolution Protocol) und ist eigentlich unterhalb von TCP/IP anzusiedeln.

Bearbeiten der ARP-Tabelle

Es kann ein Problem auftreten, wenn eine Netzwerkkarte zwischen Rechnern getauscht wurde. Da die IP-Adresse nun unter einer anderen MAC-Adresse zu finden ist, ist die ARP-Tabelle, die die Zuordnung von MAC-Adressen zu IP-Adressen enthält, nicht mehr gültig. Nach dem Tausch läuft diese IP-Adresse bei anderen Rechnern im Netz noch unter einer anderen MAC-Adresse oder umgekehrt. Die ARP-Tabelle kann unter UNIX mit dem Befehl `arp` bearbeitet werden. Mit der Option `-a` wird die Tabelle angezeigt, mit `-d` werden Einträge gelöscht, und mit `-s` kann ein Eintrag gesetzt werden. Näheres finden Sie in der Manpage von `arp`.

18.2 TCP/IP, der Standard

TCP/IP ist das Standardprotokoll für den Zugriff auf das Netzwerk. Es wurde von der Berkeley Universität seinerzeit für UNIX entwickelt. Diese enge Verbindung merkt man TCP/IP auch in anderen Umgebungen noch an.

18.2.1 Die IP-Adresse

Die Netzkennung in der IP-Adresse

Die IP-Adresse ist eine 32-Bit-Zahl, die die Netzwerkschnittstelle eines Computers im Netz eindeutig bestimmt. Der erste Teil der Nummer bezeichnet das Netz, in dem sich der Computer befindet, der zweite Teil den Computer selbst. Alle Computer, die direkt miteinander verbunden sind, gehören zum gleichen Netz und haben die gleiche Netzkennung in ihrer IP-Adresse. Wie Sie mit einem Computer kommunizieren können, der eine fremde Netzkennung hat, wird beim Thema Routing (ab Seite 469) behandelt. Ohne besondere Maßnahmen reagiert der Computer auf den Versuch, auf eine fremde Netzwerkadresse zuzugreifen, mit der Fehlermeldung »no route to host«.

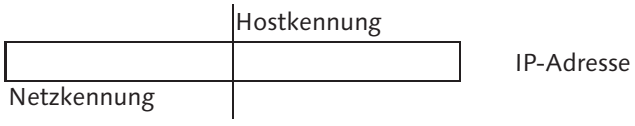

Abbildung 18.1 IP-Adresse

Die Hostkennung in der IP-Adresse

Der hintere Teil der IP-Adresse ist die Adresse des Computers im Netz. Um genau zu sein, ist sie an den Netzwerkadapter des Computers gebunden. Da aber die meisten Computer nur einen Adapter haben, soll der einfacheren Lesbarkeit halber auch weiter von »Computern« die Rede sein. Jeder Adapter muss eine eigene Nummer haben. Mehrere Rechner mit gleicher Hostkennung führen in größeren Netzen zu schwer auffindbaren Fehlern.

Darstellung

Die Grenze zwischen Netzkennung und Hostkennung liegt in lokalen Netzen oft auf einer Bytegrenze. Darum wird für IP-Adressen eine byteweise Darstellung gewählt. Da Dezimalzahlen am einfachsten zu lesen sind, schreibt man jedes der vier Byte einer IP-Adresse dezimal auf und trennt sie durch einen Punkt. Beispielsweise hat mein Arbeitsplatzrechner in meinem heimatlichen Netzwerk die IP-Adresse 192.168.109.144. Hexadezimal ergibt dies C0 A8 6D 90. Wieder als ganze Zahl geschrieben, lautet die Nummer 3232263568. Mein Macintosh hat die Adresse 192.168.109.25 oder hexadezimal C0 A8 6D 19 und damit 3232263449.

Darstellung	Arbeitsplatz	Macintosh
dezimal	3232263568	3232263449
hexadezimal	C0 A8 6D 90	C0 A8 6D 19
dotted decimal	192.168.109.144	192.168.109.25

Tabelle 18.1 Verschiedene Darstellungen von IP-Adressen

Netzwerkklasse und Netzwerkmaske

Die IP-Adresse bezeichnet also den Rechner bzw. dessen Netzadapter und das Netzwerk, in dem sich der Rechner befindet. Im Beispiel meines Netzes ist offensichtlich 192.168.109 bei beiden Rechnern gleich. Und tatsächlich benennen die ersten drei Bytes in diesem Fall das Netz und das letzte Byte den Rechner, weil in meinem Netzwerk die Netzwerkmaske auf 255.255.255.0 eingestellt ist. Man sollte also denken, ich könnte in mein privates Netz maximal 256 Rechner einbinden, da ein Byte die Werte von 0 bis 255 annehmen kann. Allerdings sind die 0 und die 255 für andere Zwecke reserviert, sodass ich mit 254 Rechnern in meinem Arbeitszimmer-Netz auskommen muss.

Netzwerkmaske und Netzwerkklasse

Welcher Teil der IP-Adresse zum Host und welche zum Netzwerk gehört, wird durch die Netzwerkmaske bestimmt. Die Maske ergab sich traditionell aus der Netzwerkklasse, zu der die IP-Adresse gehörte. Seit der Einführung von CIDR (siehe Seite 481) kann die Netzwerkmaske frei festgelegt werden. Zur optimalen Nutzung des IP-Adressraums sind die Netzwerkklassen im Internet also de facto nicht mehr vorhanden. In lokalen Netzwerken werden die Netzwerkmasken durchaus häufig noch an den Netzwerkklassen ausgelegt, auch wenn es technisch dafür keine Gründe gibt.

Die Netzwerkklasse wird durch die ersten Bits des ersten Bytes der IP-Adresse bestimmt.

Class A

Ist das erste Bit 0, gehört die Adresse zu einem Class-A-Netz. Betrachtet man das erste Byte in binärer Darstellung, sieht sich also die einzelnen Bits an, so entscheidet das erste Bit darüber, ob die Zahl größer oder kleiner als 128 ist. Ist das erste Byte einer Netzwerkadresse kleiner oder gleich 127, gehört sie also zu einem Class-A-Netz. Class-A-Netze haben eine Netzwerkmaske von 255.0.0.0.

Warum heißt nun diese Zahl »Netzwerkmaske« und warum ist sie im ersten Byte 255? Mit der Netzwerkmaske und der UND-Verknüpfung kann man den Netzanteil einer IP-Adresse herausfiltern. UND ist ein logischer Ausdruck auf Binärebene, der genau dann 1 ergibt, wenn beide Operanden 1 sind. Um das an einem Beispiel zu demonstrieren, wird die Class-A-Netzwerkadresse 10.3.4.7 mit der Netzwerkmaske 255.0.0.0 gefiltert:

```
Adresse:
dezimal     10       3        4        7
binär       00001010 00000011 00000100 00000111

Netzwerkmaske:
dezimal     255      0        0        0
binär       11111111 00000000 00000000 00000000


Adresse     00001010 00000011 00000100 00000111
Netzwerkmaske   11111111 00000000 00000000 00000000
 UND      ------------------------------------
Ergebnis    00001010 00000000 00000000 00000000
dezimal     10       0        0        0
```

Durch die UND-Verknüpfung von Adresse und Netzwerkmaske wird der Netzanteil der Adresse »herausmaskiert«. Der Netzanteil heißt hier

10.0.0.0. Das erste Byte ist die Netzwerkadresse. Der Rest ist die Adresse des Netzadapters.

Class B

Die 192 ist binär 11000000. Alle Werte, die kleiner als 192 sind, beginnen mit der Bitkombination 10 und gehören zu einem Class-B-Netz. Die Netzwerkmaske im Class-B-Netz ist 255.255.0.0, also sind die ersten zwei Bytes der Anteil der Netzwerkadresse. Der Rest ist die Rechneradresse.

Class C

Die Nummern 224-255 im ersten Byte sind reserviert und dürfen nicht als Netzwerknummern verwendet werden. Dies sind die IP-Adressen, die mit drei Einsen beginnen. Das erste Byte in einem Class-C-Netz beginnt mit der Bitkombination 110 und liegt zwischen 192 und 223 (also auch mein Hausnetz). Es hat die Netzwerkmaske 255.255.255.0 und hat damit drei Bytes für die Netzkennung und ein Byte für die verschiedenen Computer.

Bei einem größeren Unternehmen könnte ein Class-B-Netz mit den Nummern 128-191 schon sinnvoller sein, da hier ca. 16.000 Computer im Netz sein können. Richtig viele Computer passen in ein Class-A-Netz. Allerdings gibt es von diesen nur 128.

Class D

Der Bereich der Netze über 224 hat eine Sonderstellung. So dienen die Adressen des Class-D-Netzes für Multicast-Adressen. Durch das Ansprechen einer Adresse werden mehrere Rechner gleichzeitig adressiert. Anwendung findet diese Technik bei Routern und auch bei Rendezvous, einer Technologie von Apple zum Betrieb lokaler Netze, die keine Konfiguration erfordern.

	Anzahl Bits für Netze	Anzahl der Netze	Anzahl Bits für Hosts	Anzahl der Hosts	Kennungen
Class A	7	128	24	16.777.214	1-127 (0xxxxxx)
Class B	14	16.384	16	65.534	128-191 (10xxxxxx)
Class C	21	2.097.152	8	254	192-223 (110xxxxx)
Class D					224-239 (1110xxxx)
Class E					240-255 (1111xxxx)

Tabelle 18.2 Anzahl der Hosts und Netze in den Klassen

Der Hostanteil einer Nummer darf weder nur 1 noch nur 0 sein. Wenn alle Bits in einer Host-ID auf 1 gesetzt sind, ist das die Broadcast-Adresse des Netzes. Ein Paket an diese Adresse wird von allen Rechnern des Netzes gelesen. Sind alle Bits 0, bezeichnet diese Adresse das Teilnetz. Dies wird beispielsweise beim Routing so verwendet.

Verbindungsanforderung

Wenn also zwei oder mehr Rechner in einem lokalen Netz direkt miteinander verbunden werden sollen, müssen folgende Dinge gewährleistet sein:

1. Der Netzwerkteil der IP-Adressen muss gleich sein.
2. Die Hostnummern der Rechner müssen sich unterscheiden.
3. Die Hostnummer darf weder die erste Adresse sein. Sie ist 0 und ist die Netzwerkadresse. Sie darf auch nicht die letzte Nummer des Netzwerks sein, also alle Bits auf 1 stehen haben, weil dies die Broadcast-Adresse ist.

Private IP-Adressen

Adressen für lokale Netze

Für jede Klasse wurde in RFC[5] 1597 ein Bereich von Adressen definiert, die im Internet nicht weitertransportiert werden. Eine solche Adresse sollte für lokale Netze verwendet werden, sofern keine eigenen internetfähigen IP-Adressen reserviert worden sind.[6] Anders ausgedrückt heißt das, dass diese Nummern beliebig oft in der Welt verwendet werden können. Da solche Pakete im Internet nicht weitergeleitet werden, macht es gar nichts, wenn noch jemand anderes genau die gleichen Netzwerkadressen für sein lokales Netz verwendet.

Klasse	Nummernkreis
Class A	10.0.0.0
Class B	172.16.0.0 bis 172.31.0.0
Class C	192.168.0.0 bis 192.168.255.0

Tabelle 18.3 Die privaten IP-Adressen

Freie Nummern gewährleisten Kollisionsfreiheit

Wenn eine kleine Firma für ihr internes TCP/IP-Netz IP-Nummern vergeben will, sollte sie sich unbedingt Adressen aus einem dieser Bereiche aussuchen. Da keine dieser Nummern im Internet vorkommt, gibt es auch keinen Web-, FTP- oder E-Mail-Server mit einer solchen Adresse. Damit ist immer eindeutig, ob mit einer Adresse ein Rechner im lokalen Netz oder ein Server im Internet angesprochen wird. Es können auch durch eigene Adressen keine IP-Adressen ausgeblendet werden. Ein zweiter Grund hat mit der Sicherheit vor Angriffen aus dem Internet zu tun:

5 RFC ist die Abkürzung für *Request For Comment*, übersetzt etwa: Bitte um Kommentare. Die RFC stellen so etwas wie die Norm des Internets dar.

6 Internet-fähige IP-Bereiche können von einem Internet Provider für die eigene Organisation reserviert werden. In Deutschland ist dafür das Europäische RIPE in Amsterdam zuständig (http://www.ripe.net), das hierfür aber einen ausführlichen Antrag mit Begründung des Bedarfs für einen dreijährigen Zeitraum verlangt.

Da diese Nummern im Internet nicht geroutet werden, kann ein Angreifer die Rechner des Netzes kaum direkt erreichen. Er kann die Adresse des Rechners nicht durch das Internet schleusen, da diese nach RFC 1597 dort nicht transportiert wird.

Grundeinstellungen des Netzadapters: ifconfig

IP-Adressen im laufenden Betrieb ändern

Mit dem Befehl `ifconfig` kann die IP-Adresse jeder Schnittstelle festgelegt werden. Das funktioniert auch im laufenden Betrieb. Für das Umstellen der IP-Nummer ist also unter UNIX kein Reboot erforderlich. Allerdings hängen an der IP-Nummer oft einige andere Konfigurationen. Der Aufruf von `ifconfig` hat folgende Struktur:

Der Befehl ifconfig

ifconfig *<Netzadapter> <IP-Nummer>*

`ifconfig` erwartet als ersten Parameter die Bezeichnung des Netzadapter-Devices. Dabei wird allerdings der Pfadname /dev weggelassen. Die Namen dieser Gerätedateien unterscheiden sich je nach System.

System	Interface-Name
SCO:	Je nach Hardware 3b0, wdn0 und so weiter
OS/2 ftp PCTCP:	nd0
Linux:	eth0
Solaris:	le0
FreeBSD:	ed0

Tabelle 18.4 Namen für den ersten Netzwerkadapter

loopback

Das einzige Device, das auf jeder Maschine den gleichen Namen hat, ist die Loopback-Einheit. Sie heißt lo0. Dieses Pseudo-Device verweist auf die eigene Maschine.

Soll mein Arbeitsplatzrechner auf die bereits genannte IP-Adresse umgestellt werden, lautet der Befehl unter Linux:

```
ifconfig eth0 192.168.109.144
```

Hier wird der ersten Ethernet-Karte die IP-Adresse 192.168.109.144 zugeordnet.

Schnittstelleninfo

Der Befehl `ifconfig eth0` ohne weiteren Parameter zeigt die aktuelle Einstellung des Interfaces an. Beispiel:

```
gaston:~ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:00:E8:59:88:0F
          inet addr:192.168.109.144  Bcast:192.168.109.255
          Mask:255.255.255.0
          inet6 addr: fe80::200:e8ff:fe59:880f/10 Scope:Link
          inet6 addr: fe80::e859:880f/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0
          carrier:0 collisions:0 txqueuelen:100
          Interrupt:5 Base address:0xef40
```

Wir wollen uns die wichtigsten Informationen ansehen, die dieser Befehl anzeigt. Aber Sie werden sicher schon einiges wiedererkennen. In der ersten Zeile ist unter `HWaddr` (Hardware Address) die MAC-Adresse zu sehen. Sie sehen, dass die Broadcast-Adresse der Netzwerkadresse entspricht und lediglich im Bereich der Host-Adresse alle Bits auf 1 stehen. Dadurch hat das letzte Byte den Wert 255. Die Netzwerkmaske ist erwartungsgemäß 255.255.255.0. Besonders interessant ist das Wörtchen UP. Es bedeutet, dass dieses Netzinterface aktiv ist. Sie können einen Netzadapter mit dem Befehl `ifconfig` abschalten. Sie müssen zum Abstellen den zusätzlichen Parameter `down` angeben. Zur Reaktivierung dient der Parameter `up`. Beispiel:

```
ifconfig eth0 down
```

Dies ist vor allem bei temporären Verbindungen von Bedeutung. Aber es kann auch zur Wartung ganz praktisch sein, eine Maschine schnell mal aus dem Netz nehmen zu können.

Wenn es Schwierigkeiten mit der Netzanbindung gibt, kann ein Aufruf von `ifconfig` mit der Netzschnittstelle als Parameter anzeigen, ob die Schnittstelle aktiv ist. Wenn die Netzwerkhardware nicht korrekt erkannt wird, meldet der Befehl einen IOCTL-Fehler (I/O-Control). Das ist eine Fehlermeldung des Hardwaretreibers. Wie schon im Abschnitt über die Gerätedateien (siehe Seite 278) erwähnt wurde, ruft UNIX die Treiber mit dem Systemaufruf `ioctrl()` auf, wenn es Informationen über das Gerät braucht.

Dauerhaftes Einstellen der IP-Nummer

Die IP-Nummer wird normalerweise in einer der rc- oder init-Dateien festgelegt. Dazu wird wieder der Befehl `ifconfig` verwendet. Sie können die entsprechende Stelle leicht finden, indem Sie mit dem Befehl `grep` nach dem Wort »ifconfig« suchen. Allerdings wird die IP-Nummer unter-

schiedlich konfiguriert. Auf älteren UNIX-Systemen stand die IP-Nummer direkt hinter dem Aufruf von `ifconfig`, und man hat sie dort im rc-Skript einfach geändert, wenn der Rechner auf Dauer eine andere Adresse haben sollte. Finden Sie an dieser Stelle eine IP-Nummer, können Sie sie bedenkenlos ändern. Steht hinter dem `ifconfig` eine Umgebungsvariable oder finden sich dort Kommandos zum Auslesen einer Datei, sollten Sie die eigentliche Quelle suchen und die IP-Nummer dort ändern.[7]

Bei Solaris über den Hostnamen

Bei Solaris wird die IP-Adresse über den Hostnamen definiert. Für jeden Netzadapter gibt es eine Datei, die den Namen der Maschine (genauer: des Adapters) enthält. In der Datei **/etc/hostname.le0** steht beispielsweise der Name sol. Damit erhält der Netzwerkadapter le0 den Namen sol. Zur Bestimmung der IP-Nummer wird sol in der Datei **/etc/hosts** gesucht. Eine Änderung der IP-Nummer würde also in der Datei **/etc/hosts** erfolgen.

rc.config

Unter Linux ist die Art, wie die IP-Adresse festgelegt wird, von der Distribution abhängig. SUSE verwendete dazu bis zur Version 8.0 seine Universalkonfigurationsdatei **/etc/rc.config**. Im Skript **/etc/init.d/network** finden Sie den Aufruf von `ifconfig`, der die Umgebungsvariable IFCONFIG benutzt, die in **rc.config** definiert ist.

SUSE ab 8.0

Ab der Version 8.0 verwendet SUSE das Skript **/sbin/ifup**, um die Netzadapter zu initialisieren. Es wird vom rc-Skript **/etc/init.d/network** gestartet. Die bisher in der Datei **rc.config** gesammelten Konfigurationen werden nun im Verzeichnis **/etc/sysconfig** in mehrere Dateien und Unterverzeichnisse verteilt. Darin befindet sich das Verzeichnis **network**, und darin wiederum finden Sie mehrere Dateien, die die Konfiguration der verschiedenen Netzadapter bestimmen. Alle haben das Präfix **ifcfg-**. Die Datei für den Ethernet-Adapter heißt **ifcfg-eth0** und hat beispielsweise folgenden Inhalt:

```
BOOTPROTO=none
IPADDR=192.168.109.143
NETMASK=255.255.255.0
BROADCAST=192.168.109.255
NETWORK=192.168.0.0
STARTMODE=onboot
```

Die hier gesetzten Umgebungsvariablen werden in **/sbin/ifup** verwendet, um den Ethernet-Adapter zu initialisieren.

7 Wenn Sie an dieser Stelle eine IP-Nummer direkt angeben, funktioniert das natürlich auch. Falls aber jemand anders später versucht, die Nummer auf korrekte Art zu ändern, wird er sich wundern, warum nach jedem Reboot die alte IP-Nummer wieder aktiv ist.

Red Hat

Gewisse Ähnlichkeiten zur Konfiguration unter der Linux-Distribution von Red Hat sind unverkennbar. Ebenso wie dort sind im Verzeichnis **/etc/sysconfig** die Konfigurationsdateien zu finden. Bei Red Hat heißt das Unterverzeichnis **network-scripts**, in dem Sie schließlich die Datei namens **ifcfg-eth0** finden, die einen vergleichbaren Inhalt wie bei SUSE hat.

Debian

Unter Debian werden die Interfaces in **/etc/network/interfaces** eingetragen, ein Eintrag für eine Newtzwerkkarte könnte so aussehen:

```
auto eth0
iface eth0 inet static
iface eth0 inet static
        address 192.168.109.175
        netmask 255.255.255.0
        network 192.168.109.0
        broadcast 192.168.109.255
        gateway 192.168.109.10
        dns-nameservers 192.168.109.10
        dns-search mydomain
        name Ethernet LAN-Karte 0
```

Im gleichen Verzeichnis befinden sich Verzeichnisse, deren Inhalt im Zusammenhang mit dem Netzwerkstart ausgeführt werden. Der Inhalt von **if-pre-up.d** wird vor, der von **if-up.d** wird während des Hochfahrens gestartet. Zudem werden das Verzeichnis **run** ausgeführt, wenn das Netzwerk läuft. Die Skripte in **if-down.d** werden ausgeführt wenn das Netzwerk heruntergefahren wird und die in **if-post-down.d** danach.

FreeBSD

FreeBSD verwendet die Datei **rc.config**, um eine Umgebungsvariable mit den Parametern des `ifconfig` zu setzen, die beim Booten in der Datei **rc.network** als Parameter für den Aufruf von `ifconfig` verwendet wird.

Auf einer unbekannten Maschine beginnt die Suche nach der Stelle, an der die IP-Nummer festgelegt wird, an dem Ort, an dem in den rc-Dateien der Befehl `ifconfig` abgesetzt wird. Dort müssen Sie ermitteln, woher `ifconfig` seine Informationen bekommt.

Die etwas einfachere Variante dürfte die Verwendung des jeweiligen Administrationstools sein, das auf allen Plattformen eine Möglichkeit zur Einstellung der IP-Nummern anbietet. Eine kurze Übersicht über diese Programme finden Sie ab Seite 217. Die Tools haben den Vorteil, dass sie eine systemkonforme Einstellung der IP-Nummer gewährleisten.

18.2.2 Das Prüftool ping

ping prüft Verbindungen

Wenn es einen heimlichen Superstar unter den einfachen TCP/IP-Werkzeugen für die Konfiguration von Netzen gibt, so ist es `ping`. Das Programm sendet kleine Pakete an den angegebenen Zielrechner. Da dieser ein Ping-Paket sofort zurücksendet, kann Ping feststellen, ob der Rechner gestartet ist, ob die Verkabelung korrekt ist und wie schnell und zuverlässig die Verbindung zu diesem Rechner ist.

Der Befehl ping

ping [<*Optionen*>] <*IP-Nummer*>

Lokales TCP/IP prüfen

Bevor Sie fremde Rechner anpingen, sollten Sie zunächst die Nummer 127.0.0.1 aufrufen. Das ist die logische localhost-Adresse. Wenn sie versagt, funktioniert das TCP/IP auf Ihrem Rechner nicht. Anschließend sollten Sie die eigene Nummer einmal per `ping` aufrufen. Damit testen Sie, ob das eigene Netzwerkinterface richtig konfiguriert ist. Wenn das nicht klappt, werden Sie auch keine Verbindung nach außen bekommen. Schließlich eignet er sich auch zum Test der Namensauflösung (siehe Seite 490), indem Sie in mit dem Namen des Hosts aufrufen. Hat alles andere geklappt, aber der Ping erreicht nicht den Namen, stimmt die Namensauflösung nicht.

ping zum Dauerläufer machen

Bei einigen Implementationen wird nur die lapidare Meldung »xxx is alive« (beispielsweise SunOS) ausgegeben. Andere Implementierungen (so beispielsweise MS Windows) senden fünf Pakete und enden dann bereits mit einer Zusammenfassung. Beides ist unzureichend, wenn Sie beispielsweise sehen wollen, ob ein Wackelkontakt im Kabel ist. Durch entsprechende Optionen (`-t`) können Sie aber die typische Ausgabe von `ping` erhalten. Diese läuft durch, bis Sie **ctrl+C** drücken, und zeigt dann eine Statistik an:

```
PING gaston.willemer.edu (192.168.109.144): 56 data bytes
64 bytes from 192.168.109.144: icmp_seq=0 ttl=255 time=0.172 ms
64 bytes from 192.168.109.144: icmp_seq=1 ttl=255 time=0.099 ms
64 bytes from 192.168.109.144: icmp_seq=2 ttl=255 time=0.095 ms
64 bytes from 192.168.109.144: icmp_seq=3 ttl=255 time=0.093 ms
64 bytes from 192.168.109.144: icmp_seq=4 ttl=255 time=0.094 ms
64 bytes from 192.168.109.144: icmp_seq=5 ttl=255 time=0.093 ms
64 bytes from 192.168.109.144: icmp_seq=6 ttl=255 time=0.093 ms
64 bytes from 192.168.109.144: icmp_seq=7 ttl=255 time=0.098 ms
--- gaston.willemer.edu ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0.093/0.104/0.172 ms
```

Das Beispiel zeigt die Statistik einer gut laufenden Verbindung. Jedes zurückkommende Paket wird gemeldet. Die `icmp_seq` ist lückenlos. Bei dieser Nummer handelt es sich um Paketnummern. Da das Protokoll ICMP, unter dem `ping` läuft, verlorene Pakete nicht wiederholt, deutet eine durchgängige `icmp_seq` darauf hin, dass die Verbindung zuverlässig ist. Die Laufzeiten sind minimal. Dass das erste Paket etwas länger braucht, ist völlig normal.

ttl

Die Abkürzung ttl bedeutet *time to live*, also übersetzt etwa Lebensdauer. Ein Ping-Paket versucht sein Ziel auch durch Wechsel des Netzes zu erreichen. Dabei passiert es jedes Mal einen Router. Das ist ein Rechner, der die Verbindung zwischen zwei Netzwerken herstellt.

Bei jedem Wechsel wird der ttl-Wert des Pakets um eins heruntergezählt. Sobald der Wert 0 ist, wird das Paket nicht mehr weitergeleitet. Damit wird verhindert, dass Pakete, die ihr Ziel nicht finden, endlos im Internet umherschwirren. Mit der Option `-t` kann der Startwert für ttl verändert werden.

ping kann auch große Pakete senden

Sie können die Paketgröße von `ping` beliebig festlegen. Dazu gibt es je nach System den Parameter `-s`. In manchen Fällen wird die Größe in Byte als weiterer Parameter hinter dem Ziel angegeben. Mit dieser Option können Sie große Datenpakete und damit eine große Netzlast erzeugen.

Das folgende Beispiel zeigt die Ausgabe von `ping` mit einer Paketgröße von 40.000 Byte über eine nicht sehr verlässliche Verbindung:

```
gaston> ping -s 40000 192.168.109.137
PING 192.168.109.137 (192.168.109.137) from 192.168.109.144
 : 40000(40028) bytes of data.
40008 bytes from 192.168.109.137: icmp_seq=6 ttl=255 time=73.689 msec
40008 bytes from 192.168.109.137: icmp_seq=13 ttl=255 time=73.742 msec
40008 bytes from 192.168.109.137: icmp_seq=21 ttl=255 time=73.624 msec
40008 bytes from 192.168.109.137: icmp_seq=23 ttl=255 time=72.995 msec
40008 bytes from 192.168.109.137: icmp_seq=38 ttl=255 time=73.151 msec
40008 bytes from 192.168.109.137: icmp_seq=39 ttl=255 time=72.456 msec
40008 bytes from 192.168.109.137: icmp_seq=40 ttl=255 time=72.279 msec

--- 192.168.109.137 ping statistics ---
46 packets transmitted, 7 received, 84% loss, time 45090ms
rtt min/avg/max/mdev = 72.279/73.133/73.742/0.639 ms
gaston>
```

Der Wert für `time` ist recht konstant, aber natürlich höher als beim vorigen Versuch. Das liegt daran, dass die größeren Pakete länger unterwegs sind. An den großen Lücken bei `icmp_seq` ist aber zu erkennen, dass immer wieder Pakete zerstört werden. Bei einem lokalen Netzwerk kann dies ein Hinweis auf einen Wackelkontakt, einen fehlenden Abschlusswiderstand oder einen defekten Hub sein.

18.3 Routing: Verbindung mehrerer Netzwerke

Aus den verschiedensten Gründen werden Netzwerke in mehrere kleinere Netzwerke aufgeteilt. Aufgrund technischer Gegebenheiten ist dies notwendig, wenn beispielsweise unterschiedliche physikalische Netzträger verwendet werden, etwa Ethernet und Token-Ring.

Aus geografischen Gründen ist es notwendig, wenn eine Filiale über Modem oder ISDN angekoppelt werden soll. Ein weiterer Grund kann das Ziel sein, die Netzlast in den einzelnen Netzen zu reduzieren.

Gateways verbinden Netzwerke

Jeder Host kann von jedem Host im gleichen Netzwerk über seine IP-Adresse erreicht werden. Dagegen ist eine Internet-Adresse eines fremden Netzes oder Subnetzes nur dann erreichbar, wenn die Pakete weitervermittelt werden. Dazu dient ein Gateway.

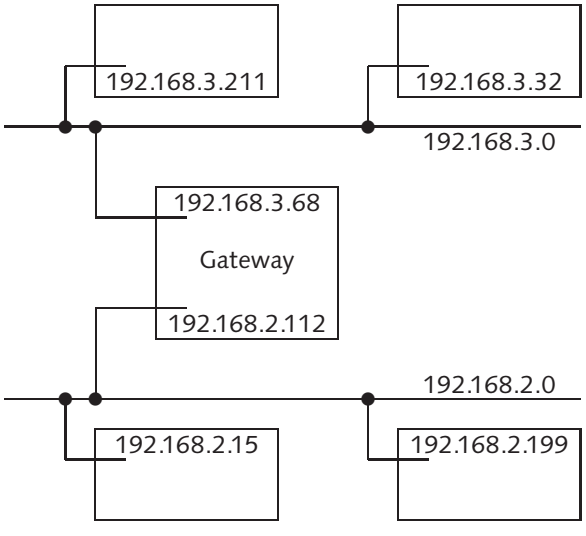

Abbildung 18.2 Ein Gateway verbindet zwei Netze

18.3.1 Gateways

Ein Gateway ist ein Computer mit zwei Netzwerkanschlüssen, die jeweils an ein anderes Netz angeschlossen sind. Jede Schnittstelle hat eine eigene IP-Adresse, die dem entsprechenden Netzwerk zugeordnet ist. Kommt ein Paket auf der einen Netzwerkkarte für das jeweils andere Netz an, wird es vom Gateway in das andere Netz eingespeist.

18.3.2 Statische Festlegung einer Route

Jeder Rechner verwaltet eine Routing-Tabelle, in der er speichert, auf welchen Wegen er welche Netze oder sogar einzelne Rechner erreichen kann. Um den Weg zu einem Rechner in einem fremden Netz zu definieren, wird der Befehl `route` verwendet. Abhängig vom Ziel gibt es drei Varianten, eine Route anzulegen:

route: Setzen einer Route

route add host <*Host*> gateway <*Gateway*> metric <*Metric*>
route add net <*Netz*> gateway <*Gateway*> metric <*Metric*>
route add default gateway *Gateway* metric *Metric*

Der erste Parameter nach `route add` ist das Ziel der Route. Mit *Gateway* wird der weiterleitende Rechner angegeben, und mit *Metric* geben Sie die Priorität der Route an.

metric

Der Parameter `metric` wird benötigt, wenn es zu einem Ziel mehrere Routen mit unterschiedlicher Geschwindigkeit gibt. Der schnellsten Route wird die höchste Priorität (beispielsweise 1) gegeben. Fällt diese aus, kann auf die Route mit der niedrigeren Priorität 2 oder 3 ausgewichen werden. Die `metric`-Information wird nur beim dynamischen Routing verwendet, um eine Bewertung der Qualität der Strecke vorzunehmen. Bei statischem Routing ist der Parameter irrelevant.[8]

Aufrufparameter von route

Leider sind die Aufrufparameter des Kommandos `route` nicht ganz einheitlich. So kann bei manchen Versionen das Schlüsselwort `gateway` auch als `gw` abgekürzt werden. Andere Implementationen benötigen die Schlüsselwörter `gateway`, `metric`, `net` und `host` gar nicht, da sie sich aus der Reihenfolge bzw. aus der Art der Parameter von selbst ergeben. Der erste Parameter ist immer das Ziel. Ob das Ziel ein Host oder ein Netz ist, lässt sich direkt an der IP-Nummer ablesen. An zweiter Stelle steht immer das Gateway, und der letzte Parameter ist immer die Priorität der Route.

route to host

Der Rechner 192.168.2.15 in Abbildung 18.2 soll als Ausgangspunkt dienen, um die Einrichtung der Routingtabelle zu demonstrieren. Wollen Sie von dort eine Route auf den Rechner 192.168.3.32 legen, muss das Paket an das Gateway gesendet werden. Die vom Rechner aus erreichbare Adresse lautet 192.168.2.112. Der Befehl dazu lautet also:

8 vgl. Hunt, Craig: TCP/IP Network Administration. O'Reilly, Sebastopol, 1994. p. 138.

```
route add host 192.168.3.32 gateway 192.168.2.112 metric 1
```

route to net

Sie brauchen nicht jeden einzelnen Rechner, sondern können mit einem Mal die Route für das gesamte Netz 192.168.3.0 angeben. Gateway und Metric bleiben gleich. Der Befehl lautet dann:

```
route add net 192.168.3.0 gateway 192.168.2.112 metric 1
```

Default-Route

Schließlich können Sie den Rechner anweisen, alle unbekannten Netzwerkadressen über ein bestimmtes Gateway hinauszuschleusen:

```
route add default gateway 192.168.2.212 metric 1
```

Der Eintrag `default` entspricht der IP-Nummer 0.0.0.0. Im Beispiel können Sie dann eine Default-Route setzen, wenn es zu anderen Netzen kein weiteres Gateway gibt. In Netzen mit Internetzugang werden normalerweise alle direkt erreichbaren Netze mit expliziten Routen bestimmt und das Gateway zum Internet auf `default` gesetzt.

Netzwerkmaske

Beim Anlegen einer Route können Sie auch die Netzwerkmaske angeben. Das ist erforderlich, wenn im lokalen Netz Subnetze (siehe Seite 477) verwendet werden. Im Internet wird durch den Einsatz von CIDR (siehe Seite 481) zu jeder Route die Netzwerkmaske angegeben. Sie geben die Netzwerkmaske durch einen zusätzlichen Parameter an, dem Sie das Schlüsselwort `netmask` voranstellen:

```
route add net 192.168.3.0 netmask 255.255.255.128 \
          gateway 192.168.2.112 metric 1
```

In diesem Fall würden die Adressen 192.168.3.1 bis 126 über das Gateway 192.168.2.112 geleitet. Der Backslash am Ende der Zeile bewirkt, dass die Befehlseingabe in der nächsten Zeile weitergeht.

Routen entfernen

Einträge in der Routing-Tabelle können wieder gelöscht werden:

route: Entfernen einer Route

route delete net <*Zieladresse*> <*Gateway*>
route delete host <*Zieladresse*> <*Gateway*>

Routen anzeigen

Der Befehl `netstat -r` zeigt die aktuellen Routingtabellen an:

```
gaston> netstat -r
Kernel IP routing table
Destination    Gateway Genmask       Iface
192.168.109.0  *       255.255.255.0 eth0
loopback       *       255.0.0.0     lo
```

In der ersten Spalte stehen die Ziele, und unter `Gateway` findet sich der Router, der aufgesucht wird, um das Ziel zu erreichen. Die Spalte `Genmask` beschreibt die Netzwerkmaske, die bei dem angegebenen Ziel vorausgesetzt wird, und zu guter Letzt folgt das Interface, über das die Pakete abgesetzt werden.

18.3.3 Statisches Routing: Ein Beispiel

[zB] Das Beispiel geht von einer Firma mit zwei Filialen und zwei Hauptgeschäftsstellen aus. Die eine Zentrale liegt in Gintoft, die andere in Norgaardholz. Jede der Zentralen hat 100 PCs. Die Filialen liegen in Hamburg und Frankfurt/Oder und haben je 2 PCs.

Die IP-Nummern für die Netze werden wie folgt festgelegt:

IP-Nummer	Bereich
192.168.108.0	Gintofter Netz
192.168.109.0	Norgaardholzer Netz
192.168.110.0	Hamburger Netz
192.168.111.0	Frankfurter Netz

Tabelle 18.5 Übersicht über die IP-Nummern im Beispiel

Die Arbeitsplätze sollen von 1 bis 190 durchnummeriert werden. Die Server sollen als 201, weitere als 202, 203 und Router als 254 angesprochen werden.

/etc/hosts Da Nummern sehr abstrakt sind, sollen die Rechner Namen erhalten. Die Arbeitsplätze in Gintoft erhalten das Präfix **gin**, die von Norgaardholz **nor**. In der Datei **/etc/hosts** (siehe Seite 491) hält UNIX die Liste, die die Namen auf die IP-Nummern abbildet. Die **/etc/hosts**-Datei sieht etwa so aus:

```
192.168.108.1     gin1
192.168.108.2     gin2
192.168.108.3     gin3
192.168.108.254   ginrout1
192.168.108.201   ginsrv1

192.168.109.1     nor1
192.168.109.2     nor2
192.168.109.3     nor3
192.168.109.254   norrout1
192.168.109.201   norsrv1
```

```
192.168.110.1     hh1
192.168.110.2     hh2
192.168.110.254   hhrout1

192.168.111.1     ffo1
192.168.111.2     ffo2
192.168.111.254   fforout1
```

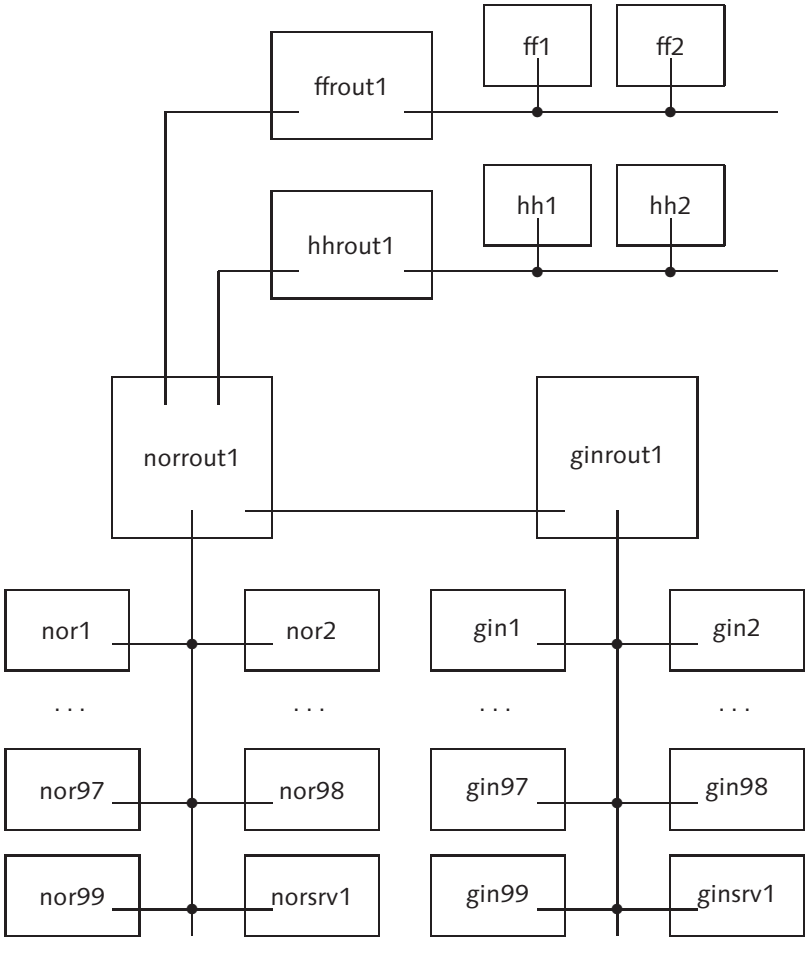

Abbildung 18.3 Netzskizze

Nur ein Gateway: Default-Route

Da es im Beispiel nur ein Gateway in jedem Netz gibt, ist die Konfiguration der Arbeitsplätze recht einfach: Sie erhalten jeweils einen Default-Eintrag auf ihren Router. Für die Norgaardholzer Arbeitsplätze und den dortigen Server lautet er:

```
route add default norrout1 1
```

route unter MS Windows

Bei Arbeitsplätzen mit MS Windows wird der Eintrag unter »Systemkonfiguration - Netzwerk - Protokolle - TCP/IP - Gateway« eingetragen. Hier ist Platz für eine oder mehrere Default-Routen. Wenn es notwendig wird, einem Windows-Rechner Routen zu verschiedenen Netzwerken über unterschiedliche Gateways zuzuweisen, erreichen Sie das nicht mehr über grafische Dialoge. Hier müssen Sie den `route`-Befehl in die **AUTOEXEC.BAT** schreiben. Wenn Sie im Beispiel davon ausgehen, dass in Norgaardholz ein Router norrout2 den Zugang zum Internet realisiert, können Sie diesen in den Eintrag »Gateway« in der Systemkonfiguration eintragen. Für die Routen zu den Netzen Gintoft, Hamburg und Frankfurt müssten Sie folgende Befehle in der **AUTOEXEC.BAT** eintragen:

```
route add 192.168.108.0 mask 255.255.255.0 norrout1
route add 192.168.110.0 mask 255.255.255.0 norrout1
route add 192.168.111.0 mask 255.255.255.0 norrout1
```

Gateway-Konfiguration zwischen Gintoft und Norgaardholz

Zwischennetz zwischen den Routern

Die Konfiguration der Router ist interessanter. norrout1 hat zwei Interfaces. Das eine mit der Nummer 192.168.109.254 ist eine Ethernet-Karte. Das andere Interface führt zur Telefonleitung und verbindet sich mit ginrout1. Zwischen ginrout1 und norrout1 gibt es also wiederum ein Netzwerk. Da dieses Netz nur den beiden Routern bekannt ist, verwenden Sie dort einfach irgendeine freie Nummer, beispielsweise 192.168.1.109 für norrout1 und 192.168.1.108 für ginrout1. Da dieses »Zwischennetz« nur zwischen den Routern bekannt ist, ist es nicht unbedingt erforderlich und wird von einigen Routern gar nicht benötigt. Allerdings wird das Verständnis des Routings durch den Wegfall auch nicht leichter. Die Routingtabelle des norrout1 lautet also:

```
# die Route auf das eigene, interne Netz
route add 192.168.109.0 192.168.109.254 1
# Gintoft über den Gintofter Router
route add 192.168.108.0 192.168.1.108 1
# Gintofter Router über WAN
route add 192.168.1.108 192.168.1.109 1
```

Damit kann norrout1 nur Nachrichten nach Gintoft bearbeiten. Andere Pakete bearbeitet er nicht. Ein Paket von Norgaardholz nach Gintoft würde nun seinen Weg finden, aber nicht mehr zurück, da in Gintoft der Router nicht weiß, wie das Paket nach Norgaardholz gesendet werden soll. Auf ginrout1 müssten die Routen also spiegelbildlich zum norrout1 eingerichtet werden. Noch einfacher wird es, wenn man davon ausgehen

kann, dass norrout1 alle Verbindungen zu den Filialen übernimmt. Dann braucht ginrout1 nur folgenden Eintrag:

```
route add default 192.168.1.109 1
```

Anbindung der Filialen

Nun werden in Norgaardholz noch die Routen zu den Filialen in Hamburg und Frankfurt/Oder gebraucht. Wir geben der WAN-Schnittstelle in Hamburg die Nummer 192.168.1.110 und der in Frankfurt die Nummer 192.168.1.111. Durch die Routen

```
route add 192.168.110.0 192.168.1.110 1  # HH über HH-Router
route add 192.168.111.0 192.168.1.111 1  # Ffo über Ffo-Router
route add 192.168.1.110 192.168.1.109 1  # HH-Router über WAN
route add 192.168.1.111 192.168.1.109 1  # Ffo-Router über WAN
```

wird eine Verbindung von Norgaardholz nach Hamburg oder Frankfurt auf gleiche Weise hergestellt wie nach Gintoft. Natürlich müssen auch hhrout1 und fforout1 mit den entsprechenden Routing-Einträgen versehen werden.

Was passiert aber, wenn Hamburg und Gintoft gleichzeitig arbeiten wollten? Damit jede Außenstelle jederzeit eine freie Leitung vorfindet, werden drei Modems oder zwei ISDN-Karten besorgt. Da jede ISDN-Karte zwei B-Kanäle betreuen kann, reichen zwei ISDN-Anschlüsse, da jeder zwei B-Kanäle besitzt. Damit die Zuordnung klar ist, werden für den norrout1 mehrere WAN-Adapter eingerichtet:

```
route add 192.168.1.110 192.168.1.2 1  # HH-Router über WAN 2
route add 192.168.1.111 192.168.1.3 1  # Ffo-Router über WAN 3
```

Alternativen

Würde nun Gintoft deutlich häufiger mit Hamburg zu tun haben als Norgaardholz, würden Sie die WAN-Anbindung dorthin natürlich nicht über Norgaardholz, sondern über Gintoft legen. Dabei würden Sie sogar einen ISDN-Anschluss sparen, da Gintoft und Norgaardholz je zwei Kanäle brauchten. Allerdings würde das Routing geringfügig komplizierter, da der Router norrout1 für eine Verbindung nach Hamburg auf ginrout1 verweisen müsste.

Insbesondere bei Firmen mit vielen Filialen stellt sich die Frage, ob man nicht abwechselnd alle Filialen über eine Telefonleitung ansprechen kann. Das hängt natürlich davon ab, wie die Filialen mit den Rechnern in den Zentralen arbeiten. Findet nur nachts ein Datenabgleich statt, kann jede Filiale einzeln nacheinander aktualisiert werden. Dazu müssen die verschiedenen Netzwerkadressen auf verschiedene Telefonnummern ab-

gebildet werden. In dem Moment, wo aber die Filialen in Konkurrenz und zeitlich nicht vorhersehbar auf die Zentralen zugreifen und umgekehrt, empfiehlt sich eine Trennung nach Kanälen. Ein ISDN-Anschluss ist nicht sehr teuer, und die Möglichkeit, dass eine Filiale die Zentrale nicht erreicht, weil die Leitung durch eine andere belegt ist, ist ausgeschlossen. Beim Thema Kosten sollte auch geprüft werden, ob es nicht sogar günstiger ist, eine Standleitung zu verwenden.

Von der IP-Nummer zur Telefonnummer

Bisher wurde völlig übergangen, wie Sie eine TCP/IP-Verbindung über ISDN legen können. Tatsächlich ist das recht einfach möglich. Leider ist die Art der Konfiguration von dem verwendeten Router abhängig.

Als Beispiel soll ein Router unter Linux verwendet werden. Dort gibt es bei jeder Distribution das Paket i4l (ISDN for Linux). Die einzelnen B-Kanäle der ISDN-Karten werden auf ISDN-Devices abgebildet. Bei einer einfachen Karte lauten sie **isdn0** und **isdn1**. Für jedes Device wird mit dem Befehl `isdnctrl` die anzurufende Telefonnummer angegeben. Sobald das Interface angesprochen wird, wird die Verbindung gewählt. Werden während einer gewissen Zeit keine Daten mehr transferiert, »legt« die Software »auf«, um Verbindungskosten zu sparen.

Sicherung des ISDN-Zugangs

Einen anrufbaren ISDN-Zugang wird jede Firma als Sicherheitsrisiko empfinden, solange nicht sichergestellt ist, dass er nur von der Filiale benutzt werden kann. Sie können für das Interface die Nummer festlegen, die der Anrufer haben muss. Alle anderen Telefonnummern werden dann durch das Interface abgewiesen. Es ist für einen Angreifer leicht, seine Telefonnummer zu unterdrücken. Aber das nützt ihm nichts, da er die Nummer der Filiale vorweisen muss, und das ist nicht so einfach.

Callback

Eine weitere Sicherungsmöglichkeit besteht darin, einen Callback zu installieren. Die Filiale ruft an und signalisiert damit, dass sie eine Verbindung haben möchte. Die Zentrale nimmt den Anruf entgegen und stellt fest, welche IP-Nummer ihr Gegenüber hat. Sie weist den Anruf jedoch zunächst zurück. Nun wählt sie die hinterlegte Telefonnummer der empfangenen IP-Nummer und ruft so die Filiale sofort zurück. Auf diese Weise ist sicher, dass nur die Filiale einen Zugang hat. Selbst wenn es einem Angreifer gelingen würde, die Telefonnummer der Filiale zu fälschen, würde die Zentrale ihn anschließend nicht zurückrufen, sondern die richtige Filiale.

Paket auf Reisen

Zur Veranschaulichung des Routings soll ein Paket von gin3 nach norsrv1 und zurück gesendet werden. Es würde also auf gin3 der Befehl

```
ping norsrv1
```

abgesetzt. Zunächst würde über die Datei **/etc/hosts** von gin3 festgestellt, dass norsrv1 die IP-Nummer 192.168.109.201 hat. Das Paket erhält diese Nummer als Zieladresse, und die eigene Adresse wird in den Absender gesteckt. Bereits dem Rechner gin3 ist klar, dass das Paket nicht zum eigenen Netz gehört. Gäbe es auf gin3 keinen Routing-Eintrag, erhielten Sie die Meldung:

gin3

```
no route to host
```

Durch die Default-Route wird das Paket erst einmal zu ginrout1 geleitet. Dieser liest wiederum die Adresse. Wäre ihm keine Route bekannt, würde er das Paket mit der Fehlermeldung »no route to host« an gin3 zurückschicken. Da er aber eine Route hat, die an die ISDN-Schnittstelle des norrout1 gerichtet ist, wird das Paket auf die Reise geschickt. norrout1 erkennt in der Adresse eine gültige Adresse für das Netz an seiner Ethernet-Karte. Also reicht er das Paket an diese Karte weiter. Auf dem Ethernet findet es nun norsrv1.

ginrout1

norrout1

Gemäß dem ping-Protokoll tauscht norsrv1 nun die Sender- und die Empfängeradresse und schickt das Paket wieder los. Würde hier nicht die Default-Route greifen, würde norsrv1 auf einem Paket sitzen, das nicht ins eigene Netz gehört und für das es keine Route gibt. Da eben gerade die Adresse von gin3 unbekannt ist, kann gin3 von diesem Problem nichts mitbekommen. Bleiben Pakete also ohne Fehlermeldung einfach aus, spricht alles dafür, dass sie korrekt das eigene Netz verlassen konnten, aber im fremden Netz nicht mehr den Weg zurück fanden. In diesem Fall können Sie sicher sein, dass das Problem auf norsrv1 oder norrout1 liegt.

nosrv1

Da es aber die Default-Route gibt, geht das Paket an norrout1. norrout1 erkennt die Nummer als Netznummer von Gintoft und besitzt eine Route dorthin über ginrout1. Dieser gibt das Paket auf das Ethernet, sodass es zu gin3 zurückfindet.

norrout1

ginrout1

18.3.4 Subnetze

Anstatt für jedes Teilnetz eine eigene Netzkennung zu verwenden, können Sie auch ein großes Netz in Unternetze zerteilen. Von außen ist diese

Subnetting ist nach außen unsichtbar

Unterteilung nicht zu sehen, und das Netz wird von fremden Routern als ein einheitliches Netz betrachtet. Erst wenn Pakete ins Innere des Netzes gelangen, werden sie durch die internen Router weitergeleitet.

Netzwerkmaske

Um ein Netzwerk derart zu unterteilen, müssen Sie die Netzwerkmaske verändern. Die Netzwerkmaske gibt an, welcher Teil der IP-Nummer das Teilnetz bestimmt und welcher Teil der Nummer einen einzelnen Rechner festlegt. Die Maske enthält für jedes Bit, das zur Netzwerkadresse gehören soll, eine 1. Für die drei Klassen von Netzen sind folgende Netzwerkmasken Standard:

Klasse	hexadezimal	dezimal
Class A	0xFF000000	255.0.0.0
Class B	0xFFFF0000	255.255.0.0
Class C	0xFFFFFF00	255.255.255.0

Tabelle 18.6 Standardnetzwerkmaske

In Abbildung 18.4 werden drei IP-Nummern der verschiedenen Netzwerkklassen anhand der Standardnetzwerkmaske in einen Netz- und einen Hostanteil aufgegliedert.

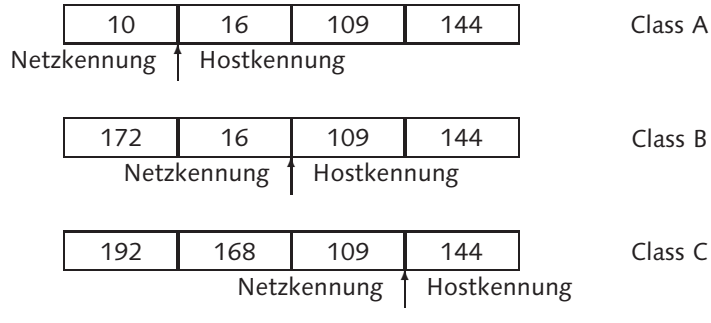

Abbildung 18.4 Netzwerkklassen und Netzwerkmaske

Die Netzwerkmaske ist bitweise konfigurierbar, und zwar kann der Anteil der Netzwerkkennung vergrößert werden. Wird eine Netzwerkmaske um ein Bit erhöht, wird damit das Netz in zwei Teilnetze zerlegt. Bei einem Class-C-Netz würde dann die Netzwerkmaske 255.255.255.128 lauten. Die 128 mag etwas überraschen. Aber von dem rechten Byte wird das am weitesten links stehende Bit verwendet. Die Dualdarstellung von 128 ist 1000000. Damit gehören alle Hostnummern, die größer als 128 sind, zu dem einen Teilnetz, und alle, die kleiner sind, gehören zu dem anderen Teilnetz.

Alternative Schreibweise

Da die Netzwerkmaske immer so aufgebaut ist, dass in der Dualdarstellung, von links beginnend, nur Einsen und ab einer gewissen Grenze dann nur noch Nullen folgen, gibt es eine alternative Schreibweise, die hinter einem Schrägstrich nur angibt, wie viele Einsen die Netzwerkmaske enthält. Für ein einfaches Class-C-Netzwerk sieht das so aus: 192.168.109.144/24. Bei der obigen Netzwerkmaske 255.255.255.128 käme noch ein Bit hinzu, also 192.168.109.144/25. Man spricht auch von der CIDR-Schreibweise (siehe Seite 481).

Bei der Teilung in Subnetze gehen auch IP-Nummern verloren, die nicht mehr verwendet werden können. Denn für beide Teilnetze gilt die Regel, dass Adressen, bei denen alle Hostbits 0 oder 1 sind, nicht für die Adressierung von Rechnern verwendet werden dürfen. Das wären im Beispiel die Nummern 0, 127, 128 und 255.

In Abbildung 18.5 ist eine Class-B-Adresse mit Subnetzwerkmaske dargestellt. Die Standardnetzwerkmaske ist bei Class B 255.255.0.0. In diesem Fall soll das erste Halbbyte der Hostkennung noch mit zur Netzwerkadresse genommen werden. Von links beginnend haben die Bits den Wert 128, 64, 32 und 16. Die andere Hälfte des Bytes bleibt null, da es nicht in die Netzwerkmaske einfließen soll. Die Summe aus 128, 64, 32 und 16 ist 240. Das ist damit die Netzwerkmaske des dritten Bytes.

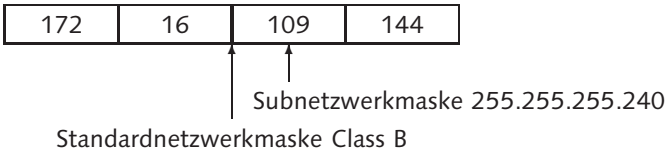

Abbildung 18.5 Subnetzwerkmaske

Netzbelastung durch Subnetting reduzieren

Eine Aufteilung des Netzes kann die Netzbelastung reduzieren. Gehen wir beispielsweise von einer Softwarefirma aus, die das Class-C-Netz mit der Nummer 192.168.2.x hat. In dem unteren Stockwerk des Firmengebäudes befindet sich die Verwaltung, deren Mitarbeiter auf einer Netzwerkplatte in unregelmäßigen Abständen Dokumente ablegen und in der Kundendatei suchen. Im oberen Stockwerk sitzen die Programmierer, die Netzwerkprogramme schreiben und von Zeit zu Zeit Belastungstests machen. Immer wenn solche Tests anlaufen, kommt die Sekretärin nicht mehr an ihre Kundendaten heran, weil das Netz überlastet ist. Schafft sie es trotzdem, werden durch ihre zusätzlichen Zugriffe die Testergebnisse verfälscht.

Man teilt das Netz daher logisch durch die Netzwerkmaske. Physisch werden die Kabel getrennt und durch ein Gateway verbunden, das zwei Netzwerkkarten besitzt. So kann ein Programmierer seine Dokumentation zwecks Rechtschreibprüfung immer noch an die Sekretärin senden. Das Gateway wird die Pakete von dem einen Netz in das andere übertragen. Der restliche Netzverkehr bleibt im jeweiligen Teilnetz.

Konsistenz der Subnetzwerkmaske

Die Netzaufteilung durch Subnetting ist nur für Rechner im Netz sichtbar, da die Maske auf den lokalen Rechnern im Netz-Device und in den Routingtabellen festgelegt wird. Es ist also wichtig, dass alle Rechner die gleiche Netzwerkmaske bekommen. Für außenstehende Rechner erscheint das Netz homogen, da sie eine Subnetzwerkmaske für ein fremdes Netz nicht kennen und so von einer Standardnetzwerkmaske ausgehen müssen.[9]

Problem bei HP-UX 10.20

In der Dokumentation von CUPS (siehe Seite 390) wird darauf hingewiesen, dass eine Subnetzdefinition, die nicht auf den Bytegrenzen liegt, bei HP-UX 10.20 Schwierigkeiten beim Broadcast verursachen könnte. In der CIDR-Schreibweise bedeutet das, dass nur 8, 16 oder 24 hinter dem Schrägstrich stehen sollte. Mit HP-UX 11 sei das Problem beseitigt.

18.3.5 Dynamisches Routen

Insbesondere das Internet mit seinen vielen Teilnetzen und der ständigen Veränderung erfordert ein Routingverfahren, das sich dynamisch anpassen kann. Die Routingtabellen werden dabei nicht vom Administrator festgelegt, sondern von Dämonen verwaltet, die mit den Dämonen der Nachbarnetze Informationen über die Qualität der Verbindung und über die Erreichbarkeit anderer Netze austauschen. Auf diese Weise werden nicht nur Engpässe oder gar Ausfälle von Leitungen erkannt. Es ist sogar möglich, die Pakete über die nächstbeste Verbindung zu schicken. In den Routingtabellen erhält der Parameter `metric` eine zentrale Bedeutung, weil er bei schlechter Verbindung erhöht wird.

Nichts für das LAN

In einem Firmennetz werden diese dynamischen Verfahren normalerweise nicht eingesetzt. Dort werden üblicherweise keine Ausfallleitungen gelegt, auf die ein dynamisches Verfahren ausweichen könnte. Dort, wo kritische Verbindungen durch Ersatzleitungen abgesichert werden, handelt es sich um ein einfaches Backup, das Sie schnell durch das manu-

9 Unter CIDR sind die Netzwerkmasken auch außerhalb des lokalen Netzes bekannt. Hier werden die Netzwerkmasken von den Routern verwaltet und weitergegeben. Aber auch hier gilt, dass ein lokales Subnetting nicht unbedingt mit der nach außen bekannten Netzwerkmaske übereinstimmen muss.

elle Ändern von zwei Routingeinträgen in Betrieb nehmen können. Die Komplexität ist überschaubar, und die Veränderungen in den Netzen sind vorhersehbar und meist gut geplant.

routed und gated

Das dynamische Routen wird beispielsweise durch den Dämon `routed` realisiert, der das RIP (Routing Information Protocol) implementiert. Der Dämon `gated` beherrscht neben RIP auch das externe Routing EGP (Exterior Gateway Protocol). Das dynamische Routen ist das Rückgrat des Internets. Da die Router ständig Informationen über die Qualität der Leitungen austauschen, können defekte Leitungen durch eine Anpassung der Routingtabellen automatisch umgangen werden.

Das EGP informiert über die Erreichbarkeit autonomer Systeme. Ein autonomes System kann ein komplexes Netzwerk mit diversen internen Routern sein. Es muss nur nach außen abgeschlossen sein.[10]

18.3.6 CIDR – Classless Inter-Domain Routing

Mit zunehmender Beliebtheit des Internets wurden dessen Engpässe immer sichtbarer. So war die Idee, 32 Bits für die IP-Nummer zu verwenden, nicht weitreichend genug, wie sich beim Ausbau des Netzes herausstellte. Zwar kann man mit 4 Byte etwa vier Milliarden Rechner durchnummerieren (und das war immerhin die damalige Größe der Weltbevölkerung), aber bei genauerem Hinsehen war die Zahl doch nicht so großzügig angesetzt. So gehen bei jedem Netz zwei Adressen für die 0 und die Broadcast-Adresse verloren. Hinzu kommt, dass eine Firma, die ein Class-C-Netzwerk betreibt, im Normalfall nicht alle 254 Adressen auch wirklich einsetzt.

RFC 1519

So wurde um 1993 das CIDR eingeführt.[11] Die Idee war, dass man jeder Netzwerkadresse eine Netzwerkmaske mitgab und dass erst durch diese bestimmt wird, wie viele Rechner in ein Teilnetz gehören. Wenn es nun noch gelingt, die benachbarten IP-Nummern lokal zu bündeln, lässt sich sogar die Anzahl der Routingeinträge reduzieren. CIDR war eigentlich als Übergangslösung bis zum Einsatz der neuen 128-Bit-IP-Adressen des IPv6 (siehe Seite 518) gedacht.

Die Änderungen durch die neue Norm konnten recht problemlos und schnell im Internet umgesetzt werden, da sie nur die Routingtabellen be-

10 vgl. Hunt, Craig: TCP/IP Network Administration. O'Reilly, Sebastopol, 1994. pp. 142.

11 vgl. Nemeth, Evi/Snyder, Garth/Seebass, Scott/Hein, Trent R.: UNIX Systemverwaltung. Markt+Technik – Prentice Hall, 2001. S. 357–360.

treffen. Oft müssen die Einstellungen nur in den Routern gesetzt werden, da die Arbeitsplätze in den meisten Fällen ohnehin nur die Defaultroute zum Gateway verwenden. Da Router ihre Routingtabellen dynamisch austauschen, ist die Konsistenz leicht zu gewährleisten. CIDR war neben dem Einsatz des Masqerading (siehe Seite 570) die entscheidende Technik, um das Problem der ausgehenden IP-Nummern zu umgehen.

In lokalen Netzwerken wird oft noch nach den alten Klassen konfiguriert, obwohl inzwischen jedes Netzwerkinterface mit einer Netzwerkmaske frei zu konfigurieren sein sollte. Im Allgemeinen wird der Administrator nicht so knappe Adressräume haben wie im Internet. Der Vorteil der alten Klassen ist, dass man die Netzwerkmasken nicht explizit dokumentieren muss, sondern sie sich von selbst verstehen.

18.4 Ohne Kabel: WLAN

WLAN ist eine beliebte Alternative zur Anbindung an ein Netzwerk über ein Ethernetkabel. Die meisten Privathäuser besitzen leider keine Netzwerkverkabelung und so ist WLAN eine praktische Möglichkeit, Computer in das Netzwerk zu integrieren, die ein paar Wände oder Stockwerke entfernt stehen.

Treiberunterstützung

Nicht alle WLAN-Adapter werden von UNIX unterstützt. Relativ problemlos ist die Installation eines Centrino-Notebooks unter Linux. Die Firmware für die WLAN-Bausteine sind im Modul ipw2100 enthalten. Für andere Hersteller finden sich die Treiber oft im Internet. Allerdings muss man dazu erst einmal herausfinden, welcher Chip in den Geräten eingebaut ist. Bei Notebooks ist dies noch vergleichsweise einfach. Bei USB-Adaptern sind diese Informationen insbesondere vor dem Kauf kaum zu ermitteln.

18.4.1 Access Point

Ein Access Point ist ein Gerät mit einem Netzwerkanschluss und einer WLAN-Antenne. Er transportiert seine Daten von einem meist ethernetbasierten LAN per Funk weiter an die mit WLAN ausgestatteten Rechner. Die WLAN-Adapter gehören in solchen Fällen zum gleichen Netzwerk wie die LAN-Adapter. Bei Netzen mit größerer Netzlast ist es sinnvoll, vor den Acess Point einen Router zu setzen, so dass das WLAN ein separates Netzwerk darstellt, in dem nur die Pakete unterwegs sind, die für die per Funk angeschlossenen Teilnehmer interessant sind.

Ad Hoc

Ein Access Point kann auch im Ad-Hoc-Modus betrieben werden. In diesem Fall ist sein Zweck nicht die Weiterleitung der Pakete an ein LAN, sondern bildet den Vermittlungspunkt zwischen mehreren Computern mit WLAN-Anschluss.

Ein Access Point ist häufig in den Internet-Routern enthalten, die von den DSL-Providern als Anschlussmöglichkeit angeboten werden. Diese Geräte bestehen meist aus einem Router ins Internet, einem kleinem Switch zum Anschluss mehrerer Ethernetkabel und einem Access Point. Meist enthalten diese Geräte auch gleich noch eine Firewall und einen DHCP-Server. Auch wenn es von außen nicht immer zu erkennen ist, laufen diese Geräte oft unter Linux, das in ihrem Flashspeicher abgelegt ist.

18.4.2 Grundinformationen

Ein Netzwerk nach IEEE 802.11 wird durch eine Namenskennung, die sogenannte ESSID gekennzeichnet. Netzwerk-Adapter nach IEEE 802.11b können bis zu einer Geschwindigkeit von 11 MBit/s übertragen. Die IEEE 802.11g schafft bis zu 54 MBit/s. Allerdings muss man berücksichtigen, dass jede Wand und fast jeder metallische Gegenstand die Geschwindigkeit herabsetzt.

18.4.3 Sicherheitsaspekte

Die Signale eines WLAN werden standardmäßig unverschlüsselt versandt. Da viele Access Points auch einen DHCP-Service liefern, ist es leicht möglich, mit einem unkonfigurierten WLAN-fähigen Rechner direkt auf fremde Kosten ins Internet zu gelangen. Vielen Besitzern eines solchen Zugangs sind die Kosten durch Fremdnutzer gleich. Sie haben ihr WLAN oft im Zusammenhang mit einer Flate Rate bekommen.

Rechtliche Konsequenzen

Ein offener Zugang ins Internet ermöglicht es einem Angreifer aber auch, unter der Anschlusskennung des WLAN-Besitzers unerlaubte Dinge zu tun. So bietet es sich förmlich an, illegale Raubkopien oder pornografische Darstellungen über den Zugang des Nachbarn zu laden. Im Falle einer strafrechtlichen Verfolgung wird die Polizei an dessen Wohnungstür klingeln. Nach aktueller deutscher Rechtsprechung wird der Betrieb eines unverschlüsselten WLAN als Teilschuld bewertet, wenn ein Dritter darüber Rechtsverletzungen begeht. Im juristischen Slang wird ein solcher Betreiber als Störer bezeichnet, obwohl er die Kommunikation in

diesem Falle ja nicht stört, sondern erst ermöglicht.[12] Es ist aber leicht nachzuvollziehen, dass der Nachbar beim illegalen Herunterladen raubkopierter Musik lieber Ihren offenen WLAN-Zugang nutzen wird als seinen eigenen. Falls eines Tages die Musikindustrie an die Tür klopft, wird es aufgrund der IP-Adressenrückverfolgung Ihre Tür sein und nicht seine.

Ein weiteres Problem ist, dass mit dem WLAN ein offener Zugang zum Netzwerk vorliegt, der von außen leicht erreichbar ist. Wer zwischen Windows-Rechnern ein paar Freigaben veröffentlicht, denkt oft nicht daran, dass der WLAN-Zugang zum Internet auch das lokale Netzwerk veröffentlicht. Hinzu kommt, dass auf Windows-Rechnern standardmäßig ein paar Freigaben eingerichtet sind, von denen der normale Anwender gar nichts weiß.

Verschlüsselung per WEP

Die Daten zwischen einem Access Point und einer WLAN-Karte können verschlüsselt werden. Dazu gibt es diverse Verfahren, die alle mit einem Schlüssel arbeiten. Das WEP ermöglicht eine Verschlüsselung mit einem Schlüssel von 128 und mit 64 Bit. Dementsprechend heißen sie WEP128 (manchmal auch WEP104) oder WEP64 (alias WEP40). Die Eingabe des Schlüssels kann je nach Software als Zeichenkette oder in hexadezimaler Fassung erfolgen. Falls der Kontakt gar nicht zustande kommen will, kann der Grund darin liegen, dass der Hex-Code eingegeben wird, obwohl das Programm den Schlüssel als Zeichenkette erwartet. Theoretisch dauert das Knacken des WEP128 zwar ungeheuer lange. Man hat bei diesen Berechnungen allerdings übersehen, dass sich bestimmte Kombinationen in der Kommunikation vorhersehbar wiederholen. Dadurch ist ein Einbruch durchaus möglich, sofern der Angreifer genügend kriminelle Energie aufbringt. Die Werkzeuge dafür stehen im Internet bereit. Nach neuester Rechtsprechung sind sie natürlich verboten.[13]

WPA

Mit dem neuen Standard IEEE 802.11g wurde das Verschlüsselungsverfahren WPA eingeführt. Der Hauptunterschied zu WEP liegt darin, dass der Schlüssel alle 10 KByte geändert wird. Damit sollte dann ein Angreifer ausreichend verwirrt werden, um einen Einstieg zu vermeiden. Allerdings müssen für den Einsatz von WPA alle Geräte mit einem WLAN-Adapter nach IEEE 802.11g ausgestattet sein. Außerdem muss darauf geachtet werden, dass auch die Treibersoftware der Karten WPA beherrscht. Das Problem betrifft aber weniger Linux, weil dort die Verfügbarkeit der

12 Diese Ausführungen sind mit aller Vorsicht zu genießen. Der Autor ist kein Jurist.

13 Und da niemand verbotene Software einsetzen wird, ist es nach Gesetz auch nicht notwendig und sogar verboten, dass ich Ihnen verrate, wovor Sie sich wappnen müssen.

WPA-Verschlüsselung nicht von der Großzügigkeit des Treiberherstellers des WLAN-Adapters abhängig ist.

MAC-Adresse

Die meisten Access Points ermöglichen es, nur bestimmte Rechner zu akzeptieren. Dabei wird geprüft, ob die MAC der Netzwerkkarte mit der hinterlegten übereinstimmt. Die MAC könnte man als eine Art Seriennummer für Netzwerkadapter bezeichnen. Fremde Rechner werden abgewiesen. Dieses Verfahren ist nur in Verbindung mit einer Verschlüsselung sinnvoll, weil ein Angreifer andernfalls eine zugelassene MAC-Adresse aus dem Datenverkehr abhören kann.

Der Turnschuh

Grundsätzlich gilt, dass auch eine noch so schlechte Absicherung immer noch besser ist als gar keine. Es ist wie bei der Unterhaltung zweier Gladiatoren. »Glaubst Du wirklich, dass Du dem Löwen entkommen kannst, wenn Du Laufschuhe anziehst?« – »Nein, ich muss aber auch nicht schneller als der Löwe sein. Es reicht, wenn ich schneller als Du bin.« Sucht also jemand einen WLAN-Zugangspunkt zum Internet, wird er den unverschlüsselten nehmen, egal, wie schwach die Verschlüsselung des anderen ist.

18.4.4 Softwaresteuerung des WLAN-Adapters

Um eine schnelle Anbindung an WLANs zu erreichen, gibt es auch unter UNIX immer mehr Werkzeuge. Da der schnelle Wechsel vor allem von Anwendern durchgeführt werden wird, sind die passenden Tools oft mit einer grafischen Oberfläche ausgestattet. Ein typischer Vertreter ist der KNetworkManager des KDE. Er zeigt alle Access Points der Umgebung mit Ihren ESSIDs an.

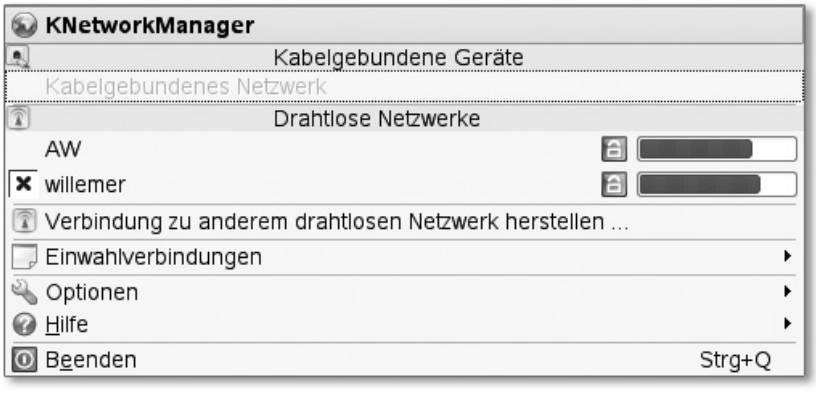

Abbildung 18.6 KNetworkManager

Die Abbildung 18.6 zeigt den KNetworkManager. In diesem Fall stehen die Netzwerke »AW« und »willemer« zur Verfügung. Mit dem letzteren ist der Rechner aktuell verbunden. Das zeigt das Kreuz. Durch Mausklick wählen Sie das Netz, in das Sie sich einwählen wollen. Bei einem bisher unbekannten Netz erscheint ein Dialog. Dort wird die Verschlüsselungsart angezeigt und es gibt die Möglichkeit, den Schlüssel anzugeben.

Konsolenzugriff

Parallel zum bekannten Befehl `ifconfig` gibt es das Programm `iwconfig`, das die Details der WLAN-Adapter anzeigt. Beispiel:[14]

```
libo:~ # iwconfig
lo        no wireless extensions.

eth0      IEEE 802.11b  ESSID:"willemer"  Nickname:"ipw2100"
          Mode:Managed  Frequency:2.462 GHz  Access Point: 00:50:FC:D4:B6:FB
          Bit Rate=11 Mb/s   Tx-Power:16 dBm
          Retry min limit:7   RTS thr:off   Fragment thr:off
          Encryption key:0010-0200-3004-0050-0600-7008-00   Security mode:open
          Power Management:off
          Link Quality=100/100  Signal level=-35 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

eth1      no wireless extensions.

sit0      no wireless extensions.

libo:~ #
```

Der aufmerksame Leser wird am Encryption key ablesen können, dass dieser WLAN-Adapter nicht WPA, sondern WEP verwendet. Das hängt damit zusammen, dass es sich um einen Adapter nach IEEE 802.11b handelt. Diese Geräte beherrschen kein WPA.

18.4.5 Treiber für WLAN-Adapter

Das Schweigen der Hersteller

WLAN-Adapter enthalten zwar einen Windows-Treiber, aber die Anwender von UNIX sind für die Hersteller offenbar noch kein Markt. Darüber hinaus geben sie auch keine Informationen an die Open-Source-Entwickler weiter, so dass es mit der Treibersituation nicht zum Besten steht. Häufig auftretende Bausteine wie der Intel-Chip im Centrino-Notebook können von Linux betrieben werden.[15] Ferner finden sich mit der Zeit immer mehr Treiber für bestimmte Chips im Internet. Die Verschwiegenheit der Anbieter geht allerdings so weit, dass der Kunden oft nicht weiß, welcher Chip in welcher Karte eingebaut ist. Es kommt sogar vor, dass

14 Den Schlüssel habe ich übrigens aus nahe liegenden Gründen geändert. Bei aller Gastfreundschaft möchte ich ungern alle Leser des Buchs in meinem Netzwerk begrüßen können.

15 Auch wenn anzumerken ist, dass dieser Treiber nicht frei ist.

in mehreren Notebooks der gleichen Typenbezeichnung unterschiedliche Chips verbaut wurden. Bei USB-Adaptern ist die Situation zur Zeit der Entstehung dieser Zeilen so, dass es überhaupt keine Treiber gibt.

Windows-Treiber

Diese Situation führt zu Verzweiflungstaten. Wenn kein Linux-Treiber verfügbar ist, wäre es immerhin schon ein Fortschritt, wenn man den Windows-Treiber verwenden kann. Genau diesen Weg beschreitet das Programm `ndiswrapper`. Es wird ausgenutzt, dass jeder Netzwerktreiber eine NDIS-Schnittstelle haben muss. Das Programm entnimmt dem Windows-Treiber diesen NDIS-Anteil und bettet ihn so ein, dass Linux diesen Treiber für das Ansprechen des Adpaters verwendet.

Die Bedienung erfolgt in zwei Schritten. Zunächst wird der Windows-Treiber in den `ndiswrapper` integriert. Im zweiten Schritt wird das Modul `ndiswrapper` geladen.

Im Beispiel wird der Siemens USB-Adapter Gigaset 108 verwendet. Auf der beiliegenden CD findet sich der Treiber in dem Verzeichnis:

/Installation/Gigaset USB Adapter 108/Driver/Windows XP & 2000

Dessen Inhalt besteht aus:

```
-r-xr-xr-x 1 arnold root 360256 27. Jul 2005  ar5523.sys
-r-xr-xr-x 1 arnold root 149392 27. Jul 2005  ar5523.bin
-r-xr-xr-x 1 arnold root  12705 22. Aug 2005  net5523.inf
-r-xr-xr-x 1 arnold root   8263 29. Aug 2005  net5523.cat
```

Diese Dateien werden kopiert und der Treiber über die INF-Datei in den `ndiswrapper` installiert.

```
libo # ndiswrapper -i net5523.inf
installing net5523 ...
libo # modprobe ndiswrapper
libo #
```

Beim nächsten Start des KNetworkManager kann bereits auf den WLAN-Adapter zugegriffen werden.

18.4.6 Funkgesteuerte Peripherie: Bluetooth

Im Gegensatz zum WLAN dient Bluetooth zur Kopplung von Geräten über kurze Distanz. Insbesondere im Mobiltelefonbereich wird diese Technik zum Anschluss von Freisprechanlagen oder zum Austausch von Adressen oder Bildern zwischen den Geräten eingesetzt.

Für PCs und Notebooks werden Bluetooth-Adapter in erster Linie zur Kommunikation mit Mobiltelefonen benutzt. Auf diese Weise können Fotos der eingebauten Kamera geladen, der Kalender synchronisiert oder als Modem für den Internetzugang von unterwegs verwendet werden.

Ein Bluetooth-Gerät hat immer eine Nummer, die aus sechs Byte besteht. Diese Nummer wird im Allgemeinen hexadezimal, durch Doppelpunkte getrennt, dargestellt.

Ist ein UNIX-Gerät mit einem Bluetooth-Adapter ausgestattet, wird der HCI-Dämon im Hintergrund gestartet, der die Kommunikationsanfragen beantwortet. Die Konfigurationsdatei **hcid.conf** finden Sie im Verzeichnis **/etc/bluetooth**. Unter den Optionen können Sie einstellen, eine Verbindungs-PIN zu hinterlegen und ob der Rechner automatisch verbinden oder den Benutzer fragen soll. Unter Device kann der Name eingestellt werden, unter dem der Computer von anderen Bluetooth-Geräten erkannt wird.

Unter KDE können Sie mit Hilfe des Konquerors eine Übersicht über alle Bluetooth-Geräte der Umgebung bekommen. Um einen Dienst zu nutzen, klicken Sie einfach das entsprechende Gerät an.

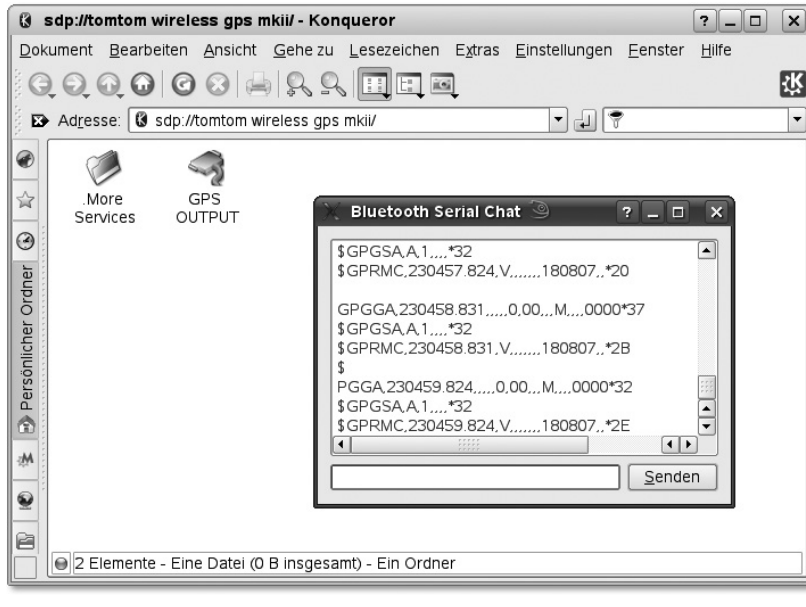

Abbildung 18.7 Bluetooth GPS Maus

Sie können sehen, welche Dienste die angeklickte GPS-Maus hat. Hier wurde der GPS OUTPUT angeklickt, und in einem Terminal werden die vom GPS-Gerät gelieferten Daten angezeigt.

In der Abbildung 18.8 sehen Sie ein Mobiltelefon, das seine Dienste anzeigt. Sie sehen auch die interne Datenstruktur, über die Sie einfach auf Dateien auf dem Telefon zugreifen können.

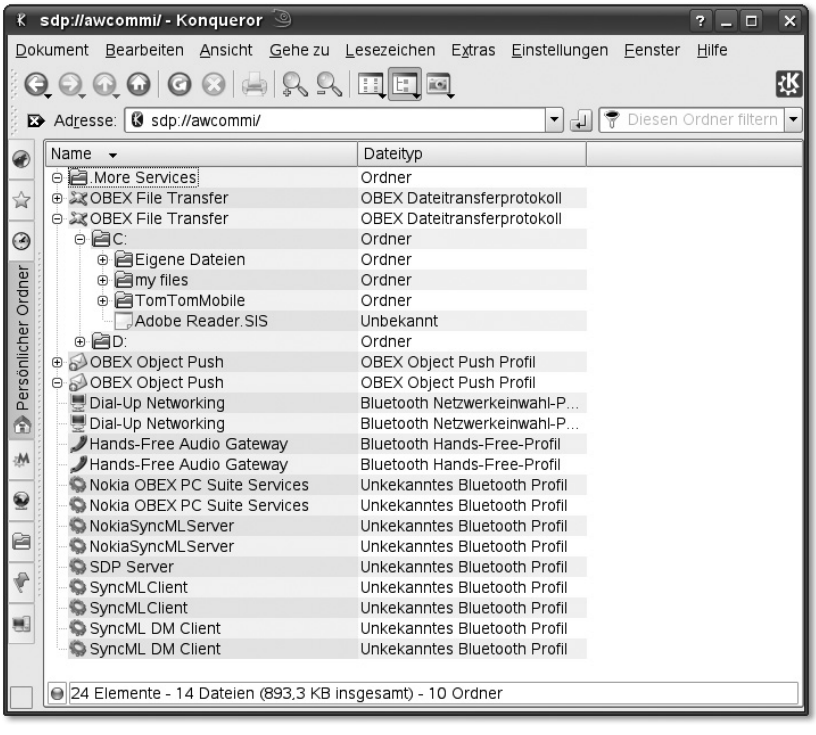

Abbildung 18.8 Bluetooth GPS Maus

18.5 Namensauflösung

Während Computer mit Zahlen wunderbar zurechtkommen, verwenden Menschen lieber Namen. Das Auflösen der Namen in Zahlen und umgekehrt kann aber wieder dem Computer übertragen werden, denn im Netzwerk ist es wichtig, dass Namen konsistent bleiben.

18.5.1 Der Host- und Domainname

Jede UNIX-Maschine hat standardmäßig einen Namen, den sogenannten Hostnamen. Dieser erhält aber erst seine besondere Bedeutung, wenn die Maschine ins Netz geht, da sie dann unter diesem Namen angesprochen werden kann. Die Zuordnung des Hostnamens zu der Maschine erfolgt durch den Befehl

hostname: Festlegen eines Rechnernamens

hostname <*Name*>

Namensgebung bei Startup

Dieser Befehl wird im Allgemeinen in einer der rc-Dateien beim Hochfahren des Systems ausgeführt. Seinen Parameter entnimmt der Befehl `hostname` oft einer besonderen Datei, beispielsweise bei Linux der Datei **/etc/HOSTNAME** oder **/etc/hostname**. Solaris besitzt sogar für jeden Netzadapter eine eigene Namensdatei, beispielsweise **/etc/hostname.le0**.

Im Internet wird der einzelne Computer vollständig mit seinem Namen und seiner Domäne identifiziert. Normalerweise wird die Domäne, durch einen Punkt abgetrennt, an den Hostnamen angehängt. Beispielsweise besagt der Name gaston.willemer.edu, dass der Hostname gaston lautet und der Rechner zur Domäne willemer.edu gehört. Innerhalb seiner Domäne braucht der Rechner nicht mit seinem Domänennamen angesprochen zu werden. Diese dreigeteilten Namen kennen Sie aus dem Internet. So ist beispielsweise bei www.willemer.de der Anteil www der Name des Rechners in der Domäne willemer.de.

Um genau zu sein, baut sich ein Domänenname von hinten nach vorn auf. So bezeichnet man das »de« als Toplevel Domain. Neben »de« für Deutschland gibt es unter anderem »nl« für Niederlande oder »dk« für Dänemark. Die Domänen in den USA wurden nicht in einer nationalen Toplevel Domain zusammengefasst, sondern werden in »com« für kommerzielle Organisationen, »gov« für Regierungsstellen und »edu« für Universitäten, Bildungs- und Forschungseinrichtungen eingeteilt. Von der Toplevel Domain wird mit dem Punkt die eigentliche Domäne abgeteilt. Diese Domänen können jeweils durch einen Punkt noch einmal in beliebig viele Subdomänen unterteilt werden. Wird mit dem Namen ein Computer bezeichnet, ist nur der erste Begriff bis zum ersten Punkt der Hostname. Beispielsweise könnte der Praktikumsrechner des Fachbereichs Informatik an der Universität Gintoft so heißen:

```
praktikum.informatik.universitaet.gintoft.de
```

18.5.2 Die Datei /etc/hosts

Um einen fremden Rechner nicht immer über seine IP-Adresse ansprechen zu müssen, gibt es die Datei **/etc/hosts**, die jeder IP-Adresse einen oder mehrere Namen zuordnet. Die Struktur eines Eintrags in der Datei **/etc/hosts** sieht folgendermaßen aus:

Struktur einer Zeile in /etc/hosts	
<ipadresse>	*<name> <nickname> . . . # <Kommentar>*

Die Reihenfolge der Namen ist wichtig

Links steht immer die IP-Nummer des Rechners. Das ist die bereits bekannte »dotted decimal«-Schreibweise, also vier Dezimalzahlen, durch Punkte getrennt. Es folgen ein oder mehrere Namen. Der erste Name ist der wichtigste. Wird eine Verbindung von außen aufgebaut, erfährt die Maschine nur die IP-Adresse. Um diesem Rechner einen Namen zu geben, wird der erste Name aus der **/etc/hosts**-Datei verwendet. Auch bei der Zuordnung von Rechten an bestimmte Rechner wird meist ein Name in die jeweilige Konfigurationsdatei eingetragen. Will ein Rechner seine Rechte geltend machen, erfährt das System seine IP-Nummer. Diese wird dann durch den ersten Namen in der **hosts**-Datei ersetzt und mit dem eingetragenen Rechnernamen verglichen. Der erste Name ist also der »offizielle« Name, die anderen Namen bezeichnet man als »nickname« (Spitzname). Nach dem Hashzeichen (#) können Sie Kommentare einfügen. Auch der Hostname der eigenen Maschine sollte in der Datei eingetragen werden, da sonst Dienstanfragen an den eigenen Host über das TCP/IP nicht erkannt werden. Bei Solaris ist es sogar unverzichtbar, da sonst die Maschine ihre IP-Nummer nicht findet.

Ein Beispiel für eine **/etc/hosts**-Datei wurde bereits im Routingbeispiel auf Seite 472 gezeigt.

Normalerweise geht die Datei hosts vor

Bei der Namensauflösung wird normalerweise zunächst in der Datei **/etc/hosts** nachgesehen. Genauer gesagt, wird die Reihenfolge in der Datei **/etc/host.conf** bzw. **/etc/nsswitch.conf** festgelegt (siehe DNS; Seite 495). Das bedeutet, dass man Namen, die durch einen zentralen Server festgelegt werden sollen, hier nicht eintragen sollte. Würde beispielsweise die Datei **/etc/hosts** folgendermaßen aussehen, dann könnte der Webserver **www.galileo.de** nicht mehr erreicht werden, weil er der lokalen IP-Adresse von gaston zugeordnet wäre:[16]

16 Das braucht Sie nicht zu bekümmern, denn der Verlag mit den fantastischen Büchern hat die Webadresse www.galileocomputing.de.

```
127.0.0.1       localhost
192.168.109.144 gaston.willemer.edu     gaston
192.168.109.144 www.galileo.de
```

Hostnamen sollte man klein schreiben

Sie machen sich übrigens keine Freunde im TCP/IP-Umfeld, wenn Sie Ihre Hostnamen in Großbuchstaben setzen. In den meisten Fällen unterscheiden die Netzprotokolle zwischen Groß- und Kleinschreibung, sodass später jeder, der mit diesen Rechnern zu tun hat, mit den Großbuchstaben hantieren muss.

18.5.3 Die Datei /etc/services

Für eine Netzwerkanfrage reicht es nicht aus, nur den Namen des Servers zu kennen. Wenn eine Anfrage an einen Netzdienst wie HTTP, FTP oder den Druckdienst gestellt wird, muss man den entsprechenden Prozess wie `httpd`, `ftpd` oder `lpd` auf diesem Rechner erreichen, der die Anfrage beantwortet.

Sockets

Um eine Verbindung zwischen zwei Prozessen aufzubauen, wird auf jeder Maschine ein Socket (dt. *Steckdose*) verwendet. Die Sockets eines Rechners sind durchnummeriert. Der Client braucht einen Socket, um über diesen später eine Antwort zu bekommen. Dazu bekommt er irgendeine Socketnummer zugeteilt, die gerade frei ist. Um einen bestimmten Dienst und damit dessen Server zu erreichen, muss der Client den Socket kontaktieren, der von dem Serverprozess betreut wird.

Für Standarddienste hat man die Nummern der Sockets festgelegt. Sie erreichen den Webserver eines Rechners im Regelfall über den Socket 80. Das Protokoll heißt HTTP (Hypertext Transfer Protocol), und der Serverprozess, der Anfragen über dieses Protokoll beantwortet, heißt httpd. In Abbildung 18.9 sehen Sie zwei Clientprozesse namens `mozilla` und `netscape`, die auf dem Rechner 192.168.109.137 laufen. Sie stellen beide Anfragen an den Webserver auf dem Rechner 192.168.109.144. Die Sockets, die die Clients verwenden, haben eine beliebige Nummer, die ihnen zufällig vom System zugeordnet wird. Sie rufen den Socket 80 auf dem Zielrechner an. Diese Nummer hat der Webserverprozess `httpd` beim Start angefordert.

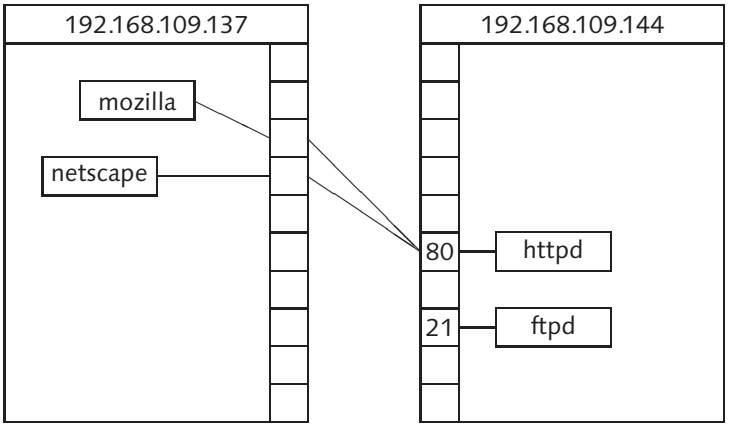

Abbildung 18.9 Kommunikation über Sockets

Im Zusammenhang mit den festgelegten Serversockets wird von einem Port gesprochen. Um den Nummern Namen zuzuordnen, gibt es die Datei **/etc/services**. Hier sehen Sie einen Ausschnitt:

```
ftp-data          20/tcp     # File Transfer [Default Data]
ftp-data          20/udp     # File Transfer [Default Data]
ftp               21/tcp     # File Transfer [Control]
telnet            23/tcp     # Telnet
telnet            23/udp     # Telnet
http              80/tcp     # World Wide Web HTTP
http              80/udp     # World Wide Web HTTP
www               80/tcp     # World Wide Web HTTP
www               80/udp     # World Wide Web HTTP
www-http          80/tcp     # World Wide Web HTTP
www-http          80/udp     # World Wide Web HTTP
```

well known port

Unter der Nummer 80 finden Sie den Dienst http und unter 21 den Dienst ftp. Und da ftp in der Regel unter der Portnummer 21 zu erreichen ist, spricht man von einem »well known port« (übersetzt etwa: »wohl bekannter Hafen«).

Die Protokolle tcp und udp

Das tcp hinter dem Schrägstrich ist die Kennung für das Protokoll. Neben tcp findet sich in der **/etc/services** noch das Protokoll udp. Es kann für dieselbe Nummer beide Protokolle nebeneinander geben. Ein weiteres Protokoll wurde im Zusammenhang mit dem Befehl `ping` bereits vorgestellt: ICMP.

TCP

TCP (Transmission Control Protocol) hat die Aufgabe, Daten sicher durch das Netzwerk zu transportieren. Dabei überprüft es die Netzpakete anhand einer Prüfnummer auf ihre Unversehrtheit. Beschädigte Pakete wer-

den neu angefordert. Geraten Pakete in der Reihenfolge durcheinander, weil sie beispielsweise unterschiedliche Wege genommen haben oder defekte Pakete neu angefordert wurden, sorgt TCP für die korrekte Reihenfolge. Den Anwendungsprogrammen gegenüber stellt es einen Datenstrom zur Verfügung. Das Programm muss die Aufteilung auf Pakete also nicht selbst durchführen. Zu guter Letzt veranlasst TCP den geregelten Verbindungsabbau.

UDP

UDP (User Datagram Protocol) ist da sehr viel einfacher. Es wird keine Verbindung aufgebaut, sondern es werden einfach Pakete versendet. UDP prüft nicht den Empfang oder die Reihenfolge der Pakete. Es wird lediglich gewährleistet, dass die Pakete unversehrt sind. Dieser fehlende Komfort kommt der Geschwindigkeit zugute. So arbeitet beispielsweise das verteilte Dateisystem NFS auf Basis von UDP.

ICMP

ICMP (Internet Control Message Protocol) ist noch einfacher. Es transportiert keine Daten, sondern Fehlermeldungen und Diagnoseinformationen. Das Protokoll wird meist unsichtbar von den Netzwerkschichten verwendet. Lediglich durch die Befehle `ping` und `traceroute` hat der Anwender mit diesem Protokoll direkt zu tun.

18.5.4 Netzgruppen: /etc/netgroup

/etc/netgroup

Netzgruppen sind eine besondere Zusammenfassung von Benutzern. Sie werden in der Datei **/etc/netgroup** definiert. Jede Netzgruppe hat einen Namen und wird durch ein oder mehrere eingeklammerte Tripel festgelegt, die aus Rechner, Benutzer und Domäne bestehen. Beispiel:

```
awfriends    (,arnold,) (gaston,,) (,,willemer.edu)
```

Dieser Eintrag bedeutet, dass die Netzgruppe awfriends aus allen Benutzern mit dem Namen arnold besteht. Hinzu kommen alle Benutzer eines Rechners gaston, ganz gleich in welcher Domain, und alle Rechner der Domäne willemer.edu.

Zu Netzgruppen werden Benutzer zusammengefasst, denen gemeinsame Rechte zugewiesen werden sollen. Beispielsweise können in den Konfigurationsdateien von NFS (siehe Seite 581) oder in den Dateien **.rhosts** der r-Kommandos (siehe Seite 549) auch Netzgruppen genannt werden. Um sie dort von Benutzern zu unterscheiden, wird ihnen das @-Zeichen vorangestellt.

18.5.5 Domain Name Service: DNS

Ersatz für /etc/hosts

Der Domain Name Service dient zur Auflösung von Host- und Domainnamen in IP-Adressen und umgekehrt. Er übernimmt damit die Aufgabe, die die Datei **/etc/hosts** lokal erfüllt. Der Vorteil von DNS liegt auf der Hand: Änderungen müssen nur an einer Stelle durchgeführt werden und gelten für alle Rechner im Netz.

DNS macht das Internet erst erreichbar

Die Bedeutung von DNS reicht aber weit über das lokale Netzwerk hinaus. Der Dreh- und Angelpunkt für einen Zugriff auf das Internet ist die Anbindung an einen DNS-Server. Sobald diese Anbindung erfolgt ist, erhalten Namen wie www.apple.de oder arnold@willemer.de erst die Zuordung zu den richtigen IP-Adressen.

Eine Domäne ist nicht unbedingt deckungsgleich mit einem physikalischen Netzwerk, sondern bezeichnet den Bereich, für den der Namensserver eingesetzt wird. Die kleine Firma, die als Beispiel für das Routing diente, hat zwar mehrere TCP/IP-Netze, wird aber sicher nur eine Domäne einrichten.

Wichtige Dateien

BIND (Berkeley Internet Name Domain) ist die wichtigste Implementation des DNS. Die Konfiguration des Clients erfolgt in **resolv.conf** und **host.conf** bzw. **nsswitch.conf**. Alle Dateien befinden sich im Verzeichnis **/etc**. Der Server verwendet eine Datei zur allgemeinen Steuerung des Dämons und zwei weitere Dateien je Domäne zur Namens- bzw. zur Nummernauflösung.

Primary und Secondary Server

Als Primary Server bezeichnet man denjenigen Namensserver, der die Autorität für die Domäne hat. Um sich gegen einen Ausfall des Primary Servers abzusichern, stellt man einen Secondary Server ins Netz. Dieser übernimmt die Namensauflösung, wenn der Primary Server ausfällt. Seine Informationen übernimmt der Secondary Server über das Netz vom Primary Server. Damit muss nur der Primary Server aktualisiert werden, wenn beispielsweise neue Rechner hinzukommen.

Cache Only Server

Als weitere Variante gibt es den Cache Only Server. Cache Only Server kopieren sich die Informationen des Primary Servers. Sie werden beispielsweise in Netzen verwendet, die zur Namensauflösung ansonsten eine Wählleitung in Anspruch nehmen müssten.

DNS-Client

Der Client wird als Resolver (engl. *resolve*: auflösen) bezeichnet. Dabei ist der Resolver kein eigenständiger Prozess, sondern ist durch eine Funktionsbibliothek quasi in den einzelnen Programmen enthalten. Al-

le Programme, die mit dem Namen von Rechnern arbeiten, verwenden den Systemaufruf `getservbyname()`, um die IP-Adresse dieses Rechners zu ermitteln (siehe Seite 968). Sobald UNIX diesen Aufruf erhält, prüft `getservbyname()` anhand der Einträge in der Datei **host.conf**, ob die Namensauflösung zuerst per **hosts** oder per DNS erfolgt. In neueren UNIX-Systemen wird auch die Datei **/etc/nsswitch.conf** verwendet. Soll der Name per DNS ermittelt werden, wird in der **resolv.conf** nachgesehen, welche DNS-Server zuständig sind. Dann wird direkt vom Programm aus der DNS-Server kontaktiert. Alle drei Dateien befinden sich im Verzeichnis **/etc**.

/etc/host.conf

In der Datei **host.conf** legt die Zeile, die mit order beginnt, fest, in welcher Reihenfolge die Namensauflösungsverfahren aufgerufen werden. Im unteren Beispiel wird zuerst in der Datei **/etc/hosts** nachgesehen und anschließend BIND verwendet. Diese Reihenfolge ist normalerweise auch sinnvoll, da der Zugriff auf die lokalen Dateien effizienter ist als die Suche im Netz.

```
order hosts bind
multi on
```

/etc/nsswitch.conf

In einigen Systemen wird die Reihenfolge der Informationen aus Diensten oder Konfigurationsdateien aus der Datei **/etc/nsswitch.conf** entnommen. Die Datei vermittelt nicht nur zwischen **/etc/hosts** und DNS, sondern bezieht auch NIS (siehe Seite 505) ein. Für nähere Informationen gibt es eine Manpage, die mit `man nsswitch.conf` aufgerufen werden kann. Der relevante Eintrag für das DNS würde so aussehen:

```
hosts:          files dns
```

Damit werden zuerst die Dateien (files) und dann der Namensdienst (dns) befragt.

/etc/resolv.conf

In der Datei **/etc/resolv.conf** wird definiert, zu welcher Domäne der Rechner gehört und welche Rechner Namensserver sind. Um meinen Arbeitsplatzrechner an den DNS-Dienst des Internets zu koppeln, reicht eine kleine **resolv.conf**. Beispiel:

```
domain     willemer.edu
nameserver 194.25.2.129    # frage den ISP
```

Eine Namensanfrage wird aufgrund der Datei **host.conf** zunächst lokal in der Datei **/etc/hosts** bedient, und alle anderen Namen werden beim DNS des Providers gesucht.

Wollen Sie das TCP/IP-Netz der Beispielfirma an das Internet anbinden, reicht es, eine etwas größere **resolv.conf** zu verwenden. Dabei wird hier angenommen, dass die Firma die Domäne firma.de hätte.

```
domain     firma.de
nameserver 192.168.108.201 # frage zuerst unseren Server
nameserver 194.25.2.129    # dann frage den ISP
```

Zuerst werden die Namen lokal geprüft. Damit gehen lokale Namensauflösungen nicht ins Internet. Erst wenn der Name hier unbekannt ist, wird der DNS-Server des Internet Service Providers (ISP) gefragt.

Um für eine vollständige Anbindung zu sorgen, müssen Sie natürlich ein geeignetes Routing einrichten. Da die meisten Firmen über keine eigenen internetfähigen IP-Adressen verfügen, muss noch ein Proxy (siehe Seite 571) oder ein Masquerading (siehe Seite 570) eingerichtet werden. Beides sind unterschiedliche Mechanismen, um über einen anderen Rechner, der eine Verbindung zum Internet hat, einen Zugang zu erhalten, obwohl der eigene Computer keine internetfähige Adresse hat.

Der DNS-Server named

/etc/named.conf

Der Serverprozess des DNS heißt `named`. Er verwaltet die Namenszuordnungen der IP-Adressen einer Zone. Eine Zone ist quasi eine Ebene einer Domäne. Die oberste Zone ist die Domäne selbst ohne ihre Subdomänen. Für jede der Subdomänen ist wiederum ein eigener DNS-Server zuständig.

Der DNS-Server verwendet mehrere Dateien zur Konfiguration. Die Ausgangsdatei heißt **/etc/named.conf**. Sie legt die Rolle des Rechners im DNS fest und definiert die Namen und Orte der weiteren Dateien. Dann gibt es für jede Zone zwei weitere Dateien. Die eine bildet die Namen auf IP-Adressen ab, und die andere nennt den Standardnamen für eine IP-Adresse.

named.conf

Die folgende Konfigurationsdatei verwaltet die Domäne willemer.edu mit der IP-Adresse 192.168.109.0 und stellt eine Verbindung zum Internet und dessen DNS-Server unter der IP-Adresse 194.25.2.129 her. Da willemer.edu keine Subdomäne besitzt, ist hier die Zone mit der Domäne identisch.

```
options {
        directory "/var/lib/named";
        forward only;
        forwarders { 194.25.2.129; };
};
```

```
zone "willemer.edu" {
        type master;
        file "willemer.edu.zone";
};

zone "109.168.192.in-addr.arpa" {
        type master;
        file "192.168.109.zone";
};

zone "localhost" {
        type master;
        file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "127.0.0.zone";
};

zone "." {
        type hint;
        file "root.hint";
};
```

options Im Bereich options werden Einstellungen vorgenommen, die den Namensserver als Ganzes betreffen. Mit dem Schlüsselwort `directory` wird festgelegt, wo sich die weiteren Konfigurationsdateien des Namensservers befinden. Für jede Zone und jedes Netzwerk wird eine eigene Datei benötigt.

Weiterleiter Mit dem Schlüsselwort `forward` wird festgelegt, ob Anfragen, die dieser Server nicht beantworten kann, an weitere DNS-Server weitergeleitet werden. Der Wert `only` könnte zwar assoziieren, dass nur Namensinformationen weitergereicht werden. Gemeint ist allerdings, dass der DNS-Server nur Kontakt mit den unter `forwarder` aufgeführten Rechnern aufnehmen soll. In der geschweiften Klammer hinter `forwarders` werden die IP-Adresse der Rechner angegeben, die angefragt werden sollen, wenn der lokale DNS-Server nicht weiter weiß. Wichtig ist das Semikolon am Ende der Nummern. Soll das lokale Netz gar nicht mit dem Internet oder einem anderen Netz verbunden werden, sind die beiden Einträge nicht erforderlich.

Zonen

Die erste Zone wird für die Domäne **willemer.edu** definiert. Mit dem Schlüsselwort `type` wird festgelegt, dass dieser Server der Master der Domäne ist. Hinter dem Schlüsselwort `file` wird der Dateiname festgelegt, in dem sich die Zuordnung der Namen zu den IP-Adressen befindet. Der Name ist frei wählbar. Es erleichtert natürlich die Übersicht, wenn der Domänenname Bestandteil des Dateinamens ist.

Nummernauflösung

Sollen die Rechnernamen aus den IP-Adressen ermittelt werden, muss es eine zweite Tabelle geben. Diese Informationen werden verwendet, wenn IP-Verbindungen als Namen angezeigt werden sollen oder wenn es notwendig ist, eine Anfrage mit einem angegebenen Hostnamen zu vergleichen. Enthält die Domäne mehrere Netzwerke, wird für jedes Netzwerk eine eigene Datei angelegt.

BIND 4: named.boot

Die älteren BIND-Versionen verwendeten eine Datei namens **named.boot**. Die folgende Datei hat in etwa die gleiche Wirkung wie die oben gezeigte **named.conf**, unterscheidet sich allerdings syntaktisch sehr deutlich:

```
primary  willemer.edu             /var/named/willemer.edu.zone
primary  109.168.192.IN-ADDR.ARPA /var/named/192.168.109.zone
cache    .                        /var/named/named.root
forwarders 194.25.2.129
```

Primary Server wird ein DNS-Server genannt, der die Hostnamen für die Domäne verantwortlich verwaltet. Die erste Zeile drückt aus, dass dieser Rechner ein Primary Server für die Domäne willemer.edu ist. Die Namensinformationen der Domäne werden in der Datei **willemer.edu.zone** gepflegt. Für den umgekehrten Weg, um also aus den Nummern die Namen zu ermitteln, wird **192.168.109.zone** verwendet. Die Cachedatei dient als Basis für das Vorhalten der Namen aus dem Internet. Wie Sie an diese Datei gelangen, wird später beschrieben. Der letzte Eintrag gibt den Namensserver an, an den die Anfrage weitergeleitet werden soll, wenn der Name aus einer fremden Domäne stammt. Beispielsweise kann hier der DNS-Server des Internet Providers stehen. Ein Secondary Server hätte eine ganz ähnliche **/etc/named.boot**:

```
directory /var/named
secondary  willemer.edu            192.168.109.144  willemer.edu.zone
secondary  109.168.192.IN-ADDR.ARPA 192.168.109.144  192.168.109.zone
```

Der Unterschied besteht in der ersten Spalte der zweiten und dritten Zeile. Hier sehen Sie, dass dieser Rechner ein Secondary Server für willemer.edu ist. Ferner ist die IP-Adresse des Primary Servers angegeben. Schließlich wird definiert, in welcher lokalen Datei die Informationen vom Primary Server abgelegt werden. Die erste Zeile (`directory`) ist

lediglich eine Abkürzung, damit man den Pfadnamen nicht bei jeder Dateiangabe wiederholen muss.

secondary = slave

Ab der Version 8 wird ein Secondary Server als Slave bezeichnet. Die Bezeichnung Slave findet sich in den Konfigurationsdateien nicht mehr. Stattdessen wird eine `forward`-Option auf den Primary Server und die Option auf `forward only` gesetzt. Damit ist es funktional ein Secondary Server.

Auflösung: Name nach Nummern

Die Zuordnung der Namen und IP-Adressen erfolgt weder in der Datei **named.conf** noch in **named.boot**, sondern in den Dateien, auf die dort verwiesen wird.

Tabelle Name nach Nummer

Die folgende Datei ermittelt aus den Namen die IP-Adressen. In den Beispielen wurde diese Datei als **willemer.edu.zone** bezeichnet. Die Datei ist bei der neueren BIND-Version gleich geblieben. Lediglich die erste Zeile ist neu:

```
$TTL 2W
@       IN SOA   mail.willemer.edu.  root.mail.willemer.edu. (
                 200111303       ;serial
                 360000          ;refresh every 100 hours
                 3600            ;retry after 1 hour
                 3600000         ;expire after 1000 hours
                 360000          ;default ttl is 100 hours
                 )
;                Wer ist der zustaendige Namensserver
                 IN NS gaston.willemer.edu.

;                Wer sind die zustaendigen Mail-Server
                 IN MX 10 mail.willemer.edu.

localhost        IN A    127.0.0.1

;                Alle Rechner in der eigenen Domain
mail             IN A    192.168.109.137
asterix          IN CNAME mail
gaston           IN A    192.168.109.144
powermac         IN A    192.168.109.141
```

TTL

Neu ist, dass der TTL-Wert (*Time To Live*; engl. Lebenszeit) bereits am Anfang der Datei stehen muss. Dieser Wert gibt an, wann die Daten von einem anderen Server erneut gelesen werden müssen. In der Version 8 muss er noch nicht zwingend dort stehen; in der Version 9 wird der

Eintrag allerdings verlangt. Er befindet sich entweder am Anfang der Datei vor dem SOA-Eintrag in der folgenden Form:

```
$TTL 1W
```

Oder Sie setzen den Wert im SOA-Eintrag direkt vor das Schlüsselwort IN:

```
@     86400 IN SOA ns hostmaster (
```

Fehlt dieser Eintrag, finden Sie in den syslog-Protokollen (siehe Seite 420), also beispielsweise in der **messages**-Datei, die Meldung »no TTL specified«.

SOA

Die zweite Zeile beginnt mit einem @. Es steht für den lokalen Domänennamen. Hier könnte also auch willemer.edu stehen. Die Abkürzung `SOA` bedeutet »Start Of zone Authority«. Die so bezeichnete Zeile gibt den Standardnameserver der Domäne und die E-Mail-Adresse des Verantwortlichen an. Allerdings wird das @ in der Mail-Adresse durch einen Punkt ersetzt. gaston ist der Namensserver (`IN NS`). Der Mail-Server der Domäne ist der Rechner mail (`IN MX`), der aber auch unter dem Namen asterix im Netz agiert.

Achtung: Offene Klammer

Ein Fallstrick ist, dass bei den neuen Versionen von BIND die öffnende Klammer hinter dem SOA-Eintrag noch in der gleichen Zeile stehen muss und nicht in der Folgezeile stehen darf.

Versionsinformationen

In der Klammer werden Parameter gesetzt, die Informationen über das Update-Verhalten festlegen. Die Zeile mit dem Kommentar `serial` kann irgendeine Versionsnummer sein, die allerdings bei jeder Änderung steigen muss. Es hat sich eingebürgert, hier das Datum in der Darstellung YYYYMMDD, gefolgt von einer mehrstelligen Zahl, zu verwenden, die hochgezählt wird, wenn am selben Tag mehrere Änderungen durchgeführt werden. In diesem Fall wurde die Datei zuletzt am 13.11.2001 und an diesem Tag das dritte Mal geändert.

Die anderen Zeilen liefern folgende Informationen:

- **IN NS**
  Der zuständige Namensserver für die Domäne.
- **IN MX *Nummer***
  Der Mail-Server. Es können mehrere Server angegeben werden. Die Nummer gibt die Rangfolge an. Je kleiner die Nummer ist, desto höher ist der Rang.

- **IN A**
  definiert die IP-Adresse eines Hostnamens. Ein Hostname wird nur einmal auf eine IP-Adresse abgebildet. Weitere Namen der gleichen IP-Adresse werden per CNAME angegeben.

- **IN CNAME**
  definiert für einen Rechner, der bereits mit IN A angegeben ist, einen weiteren Namen. Man spricht hier von einem Nickname, also einem Spitznamen.

Es werden die Namen mail, gaston und powermac auf IP-Adressen abgebildet, und für mail wird der Nickname asterix festgelegt.

Bei den in dieser Datei vorkommenden Namen wird vom System der Domänenname ergänzt, wenn der Name nicht mit einem Punkt beendet wird. Darum steht hinter mail.willemer.edu ein Punkt. Taucht bei einem Test der Domännname doppelt auf (also ein Hostname wie max.willemer.edu.willemer.edu), dann fehlt sicher irgendwo ein Punkt.

Tabelle Nummer nach Name

Die Datei **192.168.109.zone** ist das Gegenstück zur Namenstabelle. Hier werden die Hostanteile der IP-Adressen auf Namen abgebildet. Bei einem Class-B-Netz würden also zwei durch einen Punkt getrennte Nummern in der linken Spalte stehen. Diese Umsetzung von Nummern auf Namen wird im Allgemeinen bei Berechtigungsprüfungen oder bei Anzeigen verbundener Rechner und Ähnlichem benötigt.

Der erste Teil der Datei entspricht der Datei, die zur Namensauflösung verwendet wird. Erst zum Ende der Datei erscheinen zuerst die Hostnummern und durch IN PTR getrennt die vollständigen Namen der Rechner.

```
$TTL 2W
@       IN SOA    mail.willemer.edu. root.mail.willemer.edu. (
                  1999062702      ;serial
                  360000          ;refresh every 100 hours
                  3600            ;retry after 1 hour
                  3600000         ;expire after 1000 hours
                  360000          ;default ttl is 100 hours
                  )
;                 Wer ist der zustaendige Namensserver
                  IN NS mail.willemer.edu.

137               IN PTR  mail.willemer.edu.
144               IN PTR  gaston.willemer.edu.
```

Testen

nslookup

Das Testen einer DNS-Konfiguration erfolgt mit dem Befehl `nslookup`. Nach dem Aufruf meldet sich `nslookup` mit dem zuständigen DNS-Router. Anschließend können Sie Hostnamen eingeben, und `nslookup` zeigt die ermittelte IP-Adresse an:

```
gaston > nslookup
Default Server:  www-proxy.KI1.srv.t-online.de
Address:  212.185.254.170

> www.willemer.de
Server:  www-proxy.KI1.srv.t-online.de
Address:  212.185.254.170

Non-authoritative answer:
Name:    www.willemer.de
Address:  212.227.118.90

> www.apple.de
Server:  www-proxy.KI1.srv.t-online.de
Address:  212.185.254.170

Non-authoritative answer:
Name:    www.germany.euro.apple.com
Address:  17.254.3.153
Aliases:  www.apple.de
> exit
```

Hier wurde in einer Internetsitzung nach den Adressen www.willemer.de und der Adresse www.apple.de gefragt. Beendet wird die Sitzung mit dem Befehl `exit`. Der wiederholt auftretende Kommentar »Non-authoritative answer« bedeutet, dass der angerufene Server (hier von T-Online) kein autorisierter DNS-Server der Domäne willemer.de oder apple.de ist.

`nslookup` kennt die Option `server`, mit der angegeben werden kann, welcher DNS-Server gefragt werden soll. Diese Möglichkeit kann genutzt werden, um festzustellen, ob der eigene DNS-Server seine Informationen auch an fremde DNS-Server exportiert. Dies wird man in privaten Netzen vermeiden wollen.

Alternative dig

Das Programm `dig` (domain information groper) wird als moderne Alternative zu `nslookup` angesehen. Im Gegensatz zu diesem hat `dig` keinen interaktiven Modus, sondern wird mit jedem Ziel einzeln aufgerufen. Dafür sind die Ergebnisse sehr viel umfassender.

```
libo> dig www.apple.de

; <<>> DiG 9.3.2 <<>> www.apple.de
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60368
;; flags:qr rd ra; QUERY:1, ANSWER:2, AUTHORITY:6, ADDITIONAL:0

;; QUESTION SECTION:
;www.apple.de.              IN   A

;; ANSWER SECTION:
www.apple.de.       53295   IN   CNAME euro-red.apple.com.
euro-red.apple.com. 53295   IN   A     17.254.3.122

;; AUTHORITY SECTION:
apple.com.          390913  IN   NS    nserver.asia.apple.com.
apple.com.          390913  IN   NS    nserver.apple.com.
apple.com.          390913  IN   NS    nserver4.apple.com.
apple.com.          390913  IN   NS    nserver3.apple.com.
apple.com.          390913  IN   NS    nserver2.apple.com.
apple.com.          390913  IN   NS    nserver.euro.apple.com.

;; Query time: 354 msec
;; SERVER: 192.168.109.1#53(192.168.109.1)
;; WHEN: Thu Sep 20 14:36:18 2007
;; MSG SIZE  rcvd: 223

libo>
```

Root-Cache-Dateien

Für einen Cache Only Server benötigen Sie eine Ausgangsdatei, in der die Adressen der DNS-Server der Rootzone verzeichnet sind. Diese Datei können Sie unter dem Namen **named.root** per ftp vom Server **rs.internic.net** mit der Adresse 198.41.0.7 aus dem Verzeichnis **domain** herunterladen. Diese Datei muss als Ausgangsdatei für den Cache verwendet werden. Im Folgenden sehen Sie ein Protokoll, das zeigt, wie man die Datei herunterlädt:

```
gaston # ftp rs.internic.net
Connected to rs.internic.net.
220-********************************************************
220-*****                                                  *****
220-***** InterNIC Public FTP Server                       *****
220-*****                                                  *****
```

```
220-***** Login with username "anonymous"                *****
220-***** You may change directories to the following:   *****
220-*****                                                *****
220-*****   domain  -  Root Domain Zone Files            *****
220-*****                                                *****
220-***** Unauthorized access to this system may         *****
220-***** result in criminal prosecution.                *****
220-*****                                                *****
220-***** All sessions established with this server are  *****
220-***** monitored and logged. Disconnect now if you do *****
220-***** not consent to having your actions monitored   *****
220-***** and logged.                                     *****
220-*****                                                *****
220-************************************************************
220-
220 FTP server ready.
Name (rs.internic.net:root): anonymous
331 Guest login ok, send your e-mail address as password.
Password:
230 User ftp logged in.  Access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd domain
250 CWD command successful.
ftp> get named.root
local: named.root remote: named.root
500 'EPSV': command not understood.
227 Passive Mode (198,41,0,6,177,154)
150 Opening BINARY mode data connection for named.root.
100% |****************************|  2769   3.91 KB/s 00:00 ETA
226 Transfer complete.
2769 bytes received in 00:00 (3.07 KB/s)
ftp> quit
221-You have transferred 2769 bytes in 1 files.
221-Total traffic for this session was 4586 bytes in 1 transfer.
221 Thank you for using the FTP service on rs.internic.net.
gaston #
```

Die Datei **named.root** enthält die DNS-Server der Rootdomain, also die letzte Instanz.

18.5.6 Network Information Service: NIS

NIS zentralisiert Konfigurationsdateien

NIS (Network Information Service) hat die Aufgabe, in einem UNIX-Netzwerk die Informationen auf allen Maschinen konsistent zu halten. NIS

verwaltet Konfigurationsdateien und stellt sie netzweit zur Verfügung. Dadurch ist NIS nur für Maschinen mit UNIX interessant, da andere Systeme beispielsweise mit der Datei **/etc/passwd** wenig anfangen können.

Die Konfigurationsdateien werden auf dem Zentralrechner, dem Master, in sogenannte Maps umgewandelt. Anschließend kann der Client seine Informationen per Netz vom NIS holen. Da die NIS-Informationen auf dem Netz nicht besonders gesichert sind, eignet sich NIS nicht für Netze mit besonderen Sicherheitsanforderungen.

Zentrales Passwort

In den meisten Fällen wird man die Passwortdatei von NIS verwalten lassen, um die Benutzerkonten der Anwender über alle Maschinen konsistent zu halten. Die Datei **/etc/passwd** wird also zentralisiert. Damit haben Sie auf allen Rechnern im Netz das gleiche Passwort und müssen es nur an einer Stelle verwalten.

Yellow Pages

Die Namen der Befehle von NIS beginnen meist mit `yp`. Das hängt damit zusammen, dass Sun NIS ursprünglich »yellow pages«, also »Gelbe Seiten«, nennen wollte. Die britische Telefongesellschaft hatte allerdings ein Markenrecht auf den Begriff, sodass Sun den Dienst in NIS umbenennen musste.

Aufbereiten der Dateien

Makefile als Konfigurationstool

Im ersten Schritt muss festgelegt werden, welche Dateien per NIS verwaltet werden. Diese Dateien werden dann in die sogenannten Maps überführt. Der zentrale Befehl dazu lautet `makedbm`. Er wird allerdings normalerweise nicht direkt aufgerufen, sondern über ein Makefile, das sich im Verzeichnis **/var/yp** befindet. Hier werden später auch die Maps liegen. Ein Makefile ist eine editierbare Datei zum Generieren bestimmter Zieldateien und wird normalerweise in der Programmierung verwendet (siehe Seite 868).

Ändern Sie die Datei **Makefile** mit Ihrem Lieblingseditor. Sie finden hinter dem Label `all:` alle Dateien, die für das NIS umgestellt werden sollen. Mit dem Kommentarzeichen # können Sie die Dateien ausschließen, die Sie nicht mit NIS in dieser Domäne verwalten wollen:

```
# If you don't want some of these maps built, feel free to
# comment them out from this list.

#all:  passwd group hosts rpc services netid protocols \
#  netgrp mail shadow publickey networks ethers \
#  bootparams printcap amd.home auto.master auto.home \
#  auto.local passwd.adjunct timezone locale netmasks
all: passwd group rpc services netid
```

Nachdem **Makefile** angepasst ist, werden durch einen Aufruf von `make` mit dem Parameter `all` die Map-Dateien erstellt (auf manchen Systemen gibt es auch ein spezielles `ypmake`):

```
make all
```

Anschließend sollten im Verzeichnis **/var/yp** die Maps zu finden sein, mit deren Hilfe NIS die Anfragen später beantworten wird.

NIS-Server starten

Im nächsten Schritt wird die NIS-Datenbank für den Master initialisiert:

```
gaston# cd /var/yp
gaston# domainname willemer.edu
gaston# ypserv
gaston# /usr/lib/yp/ypinit -m

At this point, we have to construct a list of the hosts which
will run NIS servers.  gaston.willemer.edu is in the list of
NIS server hosts.  Please continue to add the names for the
other hosts, one per line.  When you are done with the
list, type a <control D>.
        next host to add:  gaston.willemer.edu
        next host to add:
The current list of NIS servers looks like this:

gaston.willemer.edu

Is this correct?  [y/n: y]  y
We need some  minutes to build the databases...
Building /var/yp/willemer.edu/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory `/var/yp/willemer.edu'
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
Updating group.bygid...
Updating rpc.byname...
Updating rpc.bynumber...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
gmake[1]: Leaving directory `/var/yp/willemer.edu'
gaston#
```

ypinit

Nach dem Start von `ypinit` fragt das Programm nach allen NIS-Servern. Der eigene Rechner wird automatisch angezeigt. Hier können Sie weitere Slave-Server eintragen, die bei einem Ausfall des Masters einspringen. Wenn keine weiteren NIS-Server in der Domäne arbeiten sollen, wird die Eingabe mit der Tastenkombination **ctrl+D** abgeschlossen.

Der Passwortserver

Damit kann dieser NIS-Server bereits Anfragen beantworten. Passwörter haben eine besondere Position. Sie sollen leicht von den Benutzern auf jedem beliebigen Client geändert werden können. Die Änderungen sollen dann sofort auf dem Masterserver aktualisiert werden. Das erledigt der zum NIS gehörige Passwortserver. Er wird wie folgt auf dem Masterserver gestartet:

```
rpc.yppasswdd -s /etc/yp/shadow -p /etc/yp/passwd -e chsh
ypserv
```

Nun sollte ein Client mit dem Aufruf von `yppasswd` sein Passwort domänenweit ändern können.

Überprüfen des Servers

NIS basiert auf dem RPC (Remote Procedure Call), der durch den Prozess `portmap` zur Verfügung gestellt wird. Dieser Prozess wird bei jedem auf RPC basierenden Server benötigt, da er die Umsetzung der Funktionsnummern durchführt. Das Programm `rpcinfo` hilft zu prüfen, ob der NIS-Server aktiv und sauber eingebunden ist:

```
gaston # rpcinfo -u localhost ypserv

 program 100004 version 1 ready and waiting
 program 100004 version 2 ready and waiting

gaston #
```

Die oben angegebene Meldung zeigt an, dass der NIS-Server korrekt in das RPC eingebunden ist und läuft.

Starten eines NIS-Clients

Zunächst muss die Domäneninformation stimmen. Dies kann durch einen Aufruf des Befehls `domainname` geprüft werden. Übrigens muss die Domäne des NIS nicht mit der Domäne des DNS übereinstimmen. Anschließend wird der Client-Dämon gestartet. Er heißt `ypbind`:

```
silver# domainname willemer.edu
silver# ypbind -broadcast
```

Diese beiden Befehle sollten in den rc-Dateien des Systems stehen, damit NIS nach dem Start bereits zur Verfügung steht. Wer Daten aus der **/etc/passwd** per NIS ermitteln will, sollte als letzten Eintrag in der lokalen **passwd**-Datei die folgende Zeile eintragen:

```
+::::::
```

Anschließend kann mit dem Befehl `ypcat` getestet werden, ob die Verteilung der **passwd**-Datei durch den NIS-Server funktioniert:

```
silver# ypbind -broadcast
silver# ypcat passwd
```

Der Befehl `ypcat` muss eine Liste von Einträgen der **passwd**-Datei liefern: Dann ist alles in Ordnung.

18.5.7 Portable Verzeichnisse LDAP

Ein Informationsdienst wie NIS ist in größeren Netzwerken allein für die zentrale Benutzerverwaltung unverzichtbar. Leider ist NIS sehr auf UNIX-spezifische Konfigurationsdateien festgelegt. In heutigen Netzwerken erwartet aber der Anwender, dass er sich mit seinem UNIX-Kennwort auch am Windows-Netz anmelden kann. Eine solche Verwaltung netzwerkweiter Daten ist mit LDAP (Lightweight Data Access Protocol) auch in hererogenen Netzwerken möglich.

Flexible Struktur

LDAP ist aber nicht auf die Benutzerverwaltung beschränkt. Es kann beliebige Daten speichern, die netzwerkweit gebraucht werden. Beispielsweise kann auch das Kundenverzeichnis einer Firma über LDAP realisiert werden. Diese Flexibilität hat dazu geführt, dass immer mehr Netzwerkdienste die Option bieten, ihre Konfigurationsdaten zumindest teilweise durch LDAP zu verwalten.

OpenLDAP

Von allen LDAP-Implementationen dürfte OpenLDAP am weitesten verbreitet sein. Aus diesem Grund wird das Vorgehen auf dieser Basis beschrieben.

Verzeichnisstruktur

In der Tabelle 18.7 sind die Objektklassen aufgeführt.

Kürzel	Objektklasse
c	Country, also Land
l	Location
o	Organisation
ou	Organisatorische Einheit
cn	Datenblatt

Tabelle 18.7 Objektklassenkürzel

Die Daten werden bei LDAP in einer Struktur abgelegt. Die Grundstruktur ist vorgegeben und unterliegt gewissen Regeln.

- Das Verzeichnis beginnt mit der Wurzel. Es existiert also immer ein Objekt namens Root.
- Unterhalb von Root liegt entweder ein Coutry (c) oder eine Organisation (o).
- Das Country-Objekt muss nicht, aber darf nur einmal existieren.
- Die Organisation muss direkt unter dem Country-Objekt stehen. Fehlt das Country, steht die Organisation direkt unter Root.
- Es darf mehrere Organisationen nebeneinander geben.
- Unterhalb einer Organisation muss ein Blatt (cn) oder eine organisatorische Einheit (ou) stehen.
- Es darf beliebig viele organisatorische Einheiten auf gleicher Ebene geben.
- Blätter dürfen nur an Organisationen (o) oder organisatorischen Einheite (ou) hängen.

Bestimmte Objektklassen kommen immer wieder vor und sind bereits vordefiniert.

Person Ein Objekt vom Typ Person hat immer einen CommonName (cn) und einen Surname (sn) und kann noch eine Description (desc), eine telephoneNumber, ein userPassword und ein seeAlso besitzen.

Ein Beispiel

Von der Wurzel ausgehend, wird eine Organisation namens MeineOrg definiert. Abteilungen sollen »Verkauf« und »Entwicklung« sein.

LDAP-Server-Konfiguration

Die grundlegende Konfiguration des Serverprozesses `slapd` erfolgt in der Datei **/etc/openldap/slapd.conf**. Eine Zeile besteht aus einem Parameter und dessen Wert.

include

Mit dem Parameter include wird eine weitere Datei in die Konfigurationsdatei hineingeladen. Das macht es möglich, die Konfiguration auf mehrere Dateien zu verteilen und etwas übersichtlicher zu halten.

```
include <DateiMitPfad>
```

Das ist auch bitter nötig. Denn mit LDAP kommt ein Rudel Schemata, die bestimmte Objekte vordefinieren, so dass nicht jeder Administrator das Rad neu erfinden muss. In den meisten Fällen gibt es auch eine Beispielkonfiguration, die einfach angepasst wird. In dieser Umgebung waren folgende Dateien standardmäßig eingebunden.

```
include    /etc/openldap/schema/core.schema
include    /etc/openldap/schema/cosine.schema
include    /etc/openldap/schema/inetorgperson.schema
include    /etc/openldap/schema/rfc2307bis.schema
include    /etc/openldap/schema/yast.schema
```

schemacheck

Der Parameter schemacheck legt fest, ob die Konfiguration sich an Richlinien halten muss (on) oder völlig frei ist (off).

```
schemacheck   off
```

database

Dieser Parameter legt fest, welches Format für die Datenbank verwendet wird. Welche Datenbank verwendet wird, hängt vom Zielsystem ab. Bei einigen Maschinen lautet der Eintrag `ldbm`, bei anderen `bdb`. Auch hier lohnt es sich, in der Beispielkonfiguration zu spicken.

directory

Der Parameter directory legt fest, in welchem Verzeichnis sich die Datenbank von LDAP befindet. In den meisten Fällen wird das **/var/lib/ldap** sein.

lastmod

lastmod legt fest, ob der LDAP-Server automatisch notiert, wer das Objekt zuletzt bearbeitet hat. Das kann erst einmal off bleiben.

suffix

Der Parameter suffix bezeichnet die Bereiche, für die der Server die Informationen halten soll. In unserem Fall wird hier die Organisation eingetragen.

```
suffix   "o=MeineOrg"
```

Für den Anfang wird ein root-Zugang eingebaut. `rootdn` bezeichnet die Benutzerkennung des Administrators, `rootpw` dessen Passwort.

```
rootdn "cn=admin, o=MeineOrg"
rootpw GehHeim
```

Zugriffe

Für die Zugriffe gibt es die Parameter defaultaccess und access. Der Parameter `defaultaccess` beschreibt, welche Rechte jemand hat, der ohne Kennung und Passwort zugreift. Der Parameter kann die Werte none, compare, search, read, write oder delete annehmen oder eine Kombination aus diesen Rechten, die durch einen senkrechten Strich getrennt werden. Der Parameter `access` kann mehrfach auftreten. Er beschreibt, wer was mit welchen Objekten tun darf.

```
access to {<Objekt>|<Attribut>}
by <Objekt> <Recht>
[ by <Objekt> <Recht> ]*
```

Hier ist die Konfigurationsdatei im Überblick:

```
include       /etc/openldap/schema/core.schema
include       /etc/openldap/schema/cosine.schema
include       /etc/openldap/schema/inetorgperson.schema
include       /etc/openldap/schema/rfc2307bis.schema
include       /etc/openldap/schema/yast.schema

schemacheck   off
database      bdb
directory     /var/lib/ldap
lastmod       off
suffix        "o=MeineOrg"
rootdn        "cn=admin, o=MeineOrg"
rootpw        GehHeim
defaultaccess read
access  to attrs=userPassword
        by self write
        by *    none

access to dn="cn=admin,o=MeineOrg"
       by *     none

access to *
       by *     read
```

Das Passwort ist hier im Klartext. Üblicherweise wird hier ein verschlüsseltes Passwort hinterlegt. Der Aufruf von `slappasswd` erstellt Passwörter nach der RFC2307.

```
libo # slappasswd
New password:
Re-enter new password:
{SSHA}TuZRRgpwrjdI63xhs/ap4Gh/AyfvPuLZ
libo #
```

Nun wird der Eintrag entsprechend angepasst:

```
rootpw        {SSHA}TuZRRgpwrjdI63xhs/ap4Gh/AyfvPuLZ
```

Erster Test

Der LDAP-Dämon wird der Einfachheit halber über den Startskript gestartet.

```
libo# /etc/init.d/slapd start
```

Fehler

Probleme werden in den Dateien **messages** oder **syslog** im Verzeichnis **/var/log** abgelegt. Die Fehlermeldungen geben insbesondere über Fehler in der Konfigurationsdatei Auskunft.

Daten per LDIF

LDAP speichert seine Daten in Binärdateien, auf die kein direkter Zugriff besteht. Sie werden befüllt, indem LDIF-Dateien importiert werden. LDIF-Dateien können Sie mit einem normalen Editor bearbeiten. Der Aufbau eines Eintrags in der LDIF-Datei folgt dem Muster:

```
dn: <Distinguished Name>
<Attribut>: <Wert>
[ <Attribut>: <Wert> ]*
```

Nun wird eine Datei namens **MeineOrg.ldif** erzeugt, die folgenden Inhalt hat:

```
dn: o=MeineOrg
objectclass: organization
o: MeineOrg

dn: cn=admin, o=MeineOrg
objectclass: person
cn: admin
description: LDAP-Administrator
```

```
dn: cn=arnold, o=MeineOrg
objectclass: person
cn: arnold
telephonenumber: 1024
mail: arnold.willemer@gmx.de
description: Arnold Willemer
```

Diese Datei kann mit dem Befehl ldapadd in die Datenbank übernommen werden.

```
ldapadd -h libo -D "cn=admin, o=MeineOrg" -W -f MeineOrg.ldif
```

Daraufhin fragt der Befehl nach dem Passwort und fügt die Daten in die Datenbank ein.

LDAP-Client-Konfiguration

Ein LDAP-Client muss wissen, in welcher Domäne er sich bewegt und welcher Rechner für ihn zuständig ist. Diese Informationen finden die Client-Prozesse in der Datei **/etc/openldap/ldap.conf**.

```
BASE    o=MeineOrg
HOST    libo
```

ldapsearch Für die Suche nach Einträgen wird das Kommando `ldapsearch` aufgerufen. Um den Eintrag für arnold zu suchen, lautet der Befehl:

```
ldapsearch "cn=arnold"
```

Es können für die Suche Wildcards verwendet werden.

ldapdelete Einträge können durch `ldapdelete` gelöscht werden.

```
ldapdelete -D "cn=admin,o=MeineOrg" -W "cn=arnold,o=MeineOrg"
```

18.6 Dynamische IP-Adressen (DHCP)

DHCP dient wie das ältere bootp dazu, die IP-Adresse eines Netzes an einem zentralen Ort zu verwalten. DHCP ist in RFC 2131 und 2132 beschrieben.

DHCP findet seine Anwendung in großen Netzwerken, bei denen möglichst ohne Eingriffe des Administrators kurzfristig neue Rechner einge-

hängt werden sollen. Insbesondere beim Einsatz von Laptops, die man schnell ins Netz einbinden will, ist dieses Verfahren sehr hilfreich.[17]

Für den Anwender ist DHCP äußerst bequem. Der Rechner fragt beim Booten nach der Netzkonfiguration. Nichts muss eingestellt werden. Wie so oft geht auch hier die Bequemlichkeit auf Kosten der Sicherheit. So kann jeder Mitarbeiter der Firma ohne Rückfrage an die Administration sein privates Notebook in das Netzwerk einbinden.

18.6.1 DHCP-Clients

In den meisten Fällen wird der Anwender nicht direkt mit dem DHCP-Client konfrontiert. Wenn Sie Ihr Betriebssystem installieren, werden Sie meist gefragt, ob Ihr Rechner per DHCP seine Netzinformationen holen soll. Falls Sie diese Auswahl bestätigen, erledigt die Installationsroutine den Rest.

dhclient

In den Start-Skripten wird das Programm `dhclient` gestartet, das sich per Broadcast nach dem DHCP-Server umsieht. Von dort bekommt er die Informationen über die IP-Adresse, die Default-Route und den DNS-Server. Über den Aufruf von `ifconfig` (siehe Seite 463) wird die Schnittstelle konfiguriert. Mit dem Befehl `route` wird die Route gesetzt. Falls ein DNS-Server angegeben wird, wird die passende Datei **/etc/resolve.conf** erzeugt.

Die Suche nach einem DHCP-Server verzögert das Booten vor allem dann etwas, wenn kein DHCP-Server im Netzwerk existiert. Daher kann das Setzen einer festen iP-Adresse den Bootvorgang beschleunigen.

Da ein neuer Rechner über das Netz nichts weiß, kann er auch keine spezifischen Anfragen produzieren. Eine DHCP-Anfrage (DHCPDISCOVER) kann daher nur als Broadcast gesendet werden. Dieses Broadcast kann nicht einmal über die Broadcastadresse des IP-Netzes erfolgen, da sie ja ebenfalls unbekannt ist. Also wird die IP-Adresse 255.255.255.255 im IP-Header als Zieladresse angeben, die Quelladresse wird auf 0.0.0.0 gesetzt. Die IP-Adresse 255.255.255.255 wird in einem Ethernet auf die Broadcastadresse ff:ff:ff:ff:ff:ff umgesetzt. Das gleiche gilt für das die Antwort (DHCPOFFER) des Servers, das DHCPREQUEST des Clients und die endgülitige Bestätigung (DHCPACK) des Servers. Alle werden über Broadcast übertragen, was ein Netzwerk sehr stark belasten kann.

17 vgl. Nemeth/Snyder/Seebass/Hein: UNIX Systemverwaltung. Markt+Technik – Prentice Hall, München, 2001, S. 372–374, und Augouros, Konstantin: Zahlen Meister! Linux Magazin 02/02, S. 54–59.

18.6.2 DHCP-Server

/etc/dhcpd.conf

Die DHCP-Implementierung des ISC (Internet Software Consortium) gilt als Standard. Zur Konfiguration verwendet der DHCP-Server des ISC die Konfigurationsdatei **/etc/dhcpd.conf**. Der Server `dhcpd` liest die Datei beim Starten und bricht den Start ab, wenn er Fehler entdeckt.

Das folgende Beispiel zeigt den Inhalt einer **/etc/dhcpd.conf**-Datei. Der DHCP-Server verwaltet die Domäne willemer.edu.

```
option domain-name "willemer.edu";
option domain-name-servers gaston.willemer.edu;

default-lease-time 600;
max-lease-time 7200;

authoritative;

log-facility local7;

ddns-update-style ad-hoc;

subnet 192.168.109.0 netmask 255.255.255.0 {
  range 192.168.109.10 192.168.109.20;
  option routers gaston.willemer.edu;
}
```

Hinter `option domain-name` wird die eigene Domäne festgelegt. Welches die DNS-Server (siehe Seite 495) der Domäne sind, steht hinter `option domain-name-servers`. Der Namensserver ist wichtig, da der neue Rechner eine passende Bindung zwischen Namen und IP erhalten muss.

Die Ressourcen, die der DHCP-Server vergibt, sind nur geliehen. Darum spricht man hier von Leasing. Der Parameter `default-lease-time` gibt die Zeit in Sekunden an, die die Ressourcen vergeben werden. Ist der vorgegebene Zeitrahmen überschritten, muss sich der Client um eine Verlängerung bemühen. Dadurch ist gewährleistet, dass nicht Ressourcen an Rechner vergeben sind, die sich längst nicht mehr im Netz befinden.

Die gesetzte Option `authoritative` besagt, dass dieser Server der offizielle DHCP-Server des Netzes ist.

`ddns-update-style` beschreibt die Form, in der die Abstimmung mit dem DNS erfolgt, und muss zwingend in der Datei **dhcpd.conf** definiert werden.

Die Option `log-facility` betrifft den syslog-Dämon (siehe Seite 420). Damit wird festgelegt, welche Facility, also welche Quelle, die Fehler des DHCP-Servers aus Sicht des syslog haben. `local7` ist eine der frei verwendbaren Quellenangaben.

Die subnet-Definition beschreibt, dass für das Subnetz 192.168.109.0 mit der passenden Netzwerkmaske der Bereich der IP-Adresse zwischen 10 und 20 verteilt werden kann. Der Router dieses Subnetzes ist gaston.

Zentral: subnet

Nachdem die Datei **dhcpd.conf** fertig gestellt wurde, sollten Sie `dhcpd` zumindest einmal von Hand starten, um zu sehen, ob und welche Fehlermeldungen es gibt. Bei folgendem Aufruf ist alles in Ordnung:

```
gaston # dhcpd
Internet Software Consortium DHCP Server V3.0rc12
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
Wrote 0 leases to leases file.
Listening on LPF/eth0/00:00:e8:59:88:0f/192.168.109.0/24
Sending on   LPF/eth0/00:00:e8:59:88:0f/192.168.109.0/24
Sending on   Socket/fallback/fallback-net
gaston #
```

Der Client braucht keine Information darüber, welcher Rechner im Netz der DHCP-Server ist. Sobald er bootet, wird er per Broadcast im lokalen Netz nach dem passenden Server fragen.

Der DHCP-Server ist auch in der Lage, einem bootp-Client seine IP-Nummer anhand seiner MAC-Adresse zuzuteilen. Dazu wird in der Datei **dhcpd.conf** folgender Eintrag gemacht.

```
group {
    host libo {
        hardware ethernet 00:0C:F1:3C:73:70;
        fixed-address 192.168.109.117;
    }
    host libowlan {
        hardware ethernet 00:0B:5D:47:A1:37;
        fixed-address 192.168.110.117;
    }
...
}
```

Hier wurde ein Eintrag für libo und einer für libowlan vorgenommen. Weitere Rechner könnten folgen. Das deuten die drei Punkte an. Die

beiden Rechner sind dieselben Geräte, allerdings einmal über das Kabel und das andere Mal per WLAN angeschlossen.

Diese MAC-Adressen sind auf verschiedene Weise zu bestimmen. Die einfachste Methode ist, Sie fragen den Rechner mit `ifconfig` selbst.

```
libo # ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:F1:3C:73:70
          inet addr:192.168.109.112  Bcast:192.168.109.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:f1ff:fe3c:7370/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:78185 errors:2 dropped:2 overruns:0 frame:0
          TX packets:116 errors:0 dropped:3 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6778918 (6.4 Mb)  TX bytes:8396 (8.1 Kb)
          Interrupt:11 Base address:0xe000 Memory:d0201000-d0201fff

eth1      Link encap:Ethernet  HWaddr 00:0B:5D:47:A1:37
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:11 Base address:0xc000
...
```

Sie sehen hinter dem Kürzel HWaddr die MAC-Adresse. Eine andere, allerdings etwas indirektere Quelle ist die Datei **/var/lib/dhcp/dhcp.leases**. Dort finden Sie alle MAC-Adressen der Rechner, die aktuell eine Nummer zugeteilt bekommen haben.

18.7 Next Generation IPv6

Als die IP-Adressen definiert wurden, hielt man die Anzahl der Adressen für extrem großzügig. Immerhin waren sie so dimensioniert, dass bei den damaligen Bevölkerungszahlen für jeden Menschen eine IP-Adresse verfügbar war: Erst wenn jeder Mensch einen eigenen Computer besäße, der ans Internet angeschlossen wäre, würden alle Adressen aufgebraucht sein. Diese Berechnung stellte sich bald als zu einfach heraus, da die Adressen netzweise vergeben wurden und dabei immer einige ungenutzte Nummern in Reserve gehalten wurden.

Als die Adressen mit der überraschend zunehmenden Verbreitung von Computern und Internetzugängen immer knapper wurden, begann man über eine Überarbeitung der IP-Adressierung nachzudenken. So liest man heute überall, dass die nächste Generation von IP-Adressen vor der Tür stehe. Nur, wie weit sie von der Tür entfernt ist, scheint noch nicht ganz klar zu sein. In Asien hat sich die Adressierung durchgesetzt und viele Provider bieten die Adressierung bereits an.

IPv6-Adressen sind 16 statt 4 Byte lang

Eine Adresse unter IPv6 ist statt 4 Byte (wie bei dem jetzigen IPv4[18]) 16 Byte oder 128 Bit lang. Es gäbe damit etwa 3,4 * 10^{38} Adressen. Damit ergibt sich eine 24-stellige Zahl von Adressen pro Quadratmeter Erdoberfläche.[19]

Allerdings soll der Nummernumfang auch genutzt werden, um einige Dinge zu vereinfachen. Es wird immer noch eine Unterteilung in Netz- und Hostanteil der Adresse geben. Allerdings wird sie nicht wie bisher von einer Netzwerkklasse abhängig gemacht, sondern die Netzkennung wird in Zukunft konstant 45 Bit und die Hostkennung wird immer 64 Bit lang werden.

T	Netzkennung	Sub	Hostkennung
3	45 Bit	16 Bit	64 Bit

Abbildung 18.10 Aufbau einer IPv6-Adresse

Das Teilnetz wird nun in der IP-Adresse codiert (Sub) und muss nicht mehr in den Routingtabellen verwaltet werden. Auch das führt zur Vereinfachung. Hinzu kommen zu Anfang 3 Bits zur Bezeichnung des Adresstyps (T). Die 64 Bit für die einzelnen Rechner sind so groß, dass man die MAC, also die Hardwareadresse der verwendeten Adapter, darin ablegen kann.[20]

Die großen Anbieter haben die Umstellung als Prestigeobjekt betrachtet und sie bereits recht weit vorangebracht. Neue Geräte sind im Normalfall in der Lage, sofort mit IPv6 zu arbeiten. Auch die Politik, allen voran die EU, hält IPv6 für unterstützenswert. So hat die EU bereits Forschungsnetze auf IPv6 umgestellt.[21] Auch hier entsteht leicht der Eindruck, dass sich die Politiker gern als Förderer der neuesten Technologie darstellen möchten, das Engagement also mehr mit Prestige als mit dem Verständnis für die Notwendigkeit dieser Technologie zu tun hat.

Probleme beim Umstieg

Ganz ohne Probleme wird eine weltweite Umstellung allerdings nicht ablaufen. Problematischer als die großen Systeme sind die einzelnen Programme und kleinere sowie ältere Systeme. Die Umstellung wirkt sich auf sämtliche Geräte wie Computer, Router und Druckserver bis auf je-

18 Ein IPv5 gibt es nicht.
19 Nemeth/Snyder/Seebass/Hein: UNIX-Systemverwaltung. Markt+Technik – Prentice Hall, München, 2001. S. 357.
20 vgl. Nemeth/Snyder/Seebass/Hein: UNIX-Systemverwaltung. Markt+Technik – Prentice Hall, München, 2001. S. 363f.
21 vgl. Heise-Ticker vom 16.1.2004.

des einzelne Netzprogramm wie ftp oder einen Webserver aus. Da die Bitzahl der IP-Adressen geändert wird, wirkt sich die Umstellung auf alle Datenstrukturen aus, die IP-Adressen halten – bis hin zu Datenbanken, bei denen Tabellenstrukturen geändert werden müssen. Da viele Systeme nicht so schnell oder gar nicht umgestellt werden können, gibt es einen Kompatibilitätsmodus, der vermutlich nach einer Umstellung noch recht lange aktiv sein muss.

Notwendigkeit fragwürdig

Aus diesem Grund ist es nicht ungewöhnlich, dass der Markt zurückhaltend auf IPv6 reagiert. Inzwischen haben Techniken wie das Masquerading (siehe Seite 570) bewirkt, dass die heraufbeschworenen Katastrophenszenarien so schnell nicht eintreten werden. Es gibt sogar kompetente Aussagen, die besagen, dass eine Umstellung gar nicht erforderlich sei.[22] Andere Autoren, wie Simson Garfinkel,[23] erwarten, dass Asien und vermutlich auch Europa umsteigen werden, da der noch verfügbare Adressraum für Asien sehr knapp ist. Garfinkel führt aus, dass von den theoretisch möglichen vier Milliarden IPv4-Adressen etwa drei Milliarden auf amerikanische Internetprovider verteilt sind. Für China und Südkorea stünden dagegen nur 38,5 beziehungsweise 23,6 Millionen Adressen zur Verfügung. Garfinkel vermutet, dass die USA, wie schon beim Versuch des Umstiegs auf das metrische System in den Siebzigern, beim Althergebrachten bleiben könnten. Neben den bereits genannten Argumenten verweist er auf Geschwindigkeitseinbußen, die seiner Ansicht nach darauf beruhen, dass viele Netzwerkkomponenten, die heute bereits in der Hardware implementiert sind, dann durch Software emuliert werden müssten. Garfinkel verweist auf Sicherheitsprobleme, da die neue IPv6-Software nicht so ausgetestet auf den Markt kommt, wie es die IPv4-Software durch ihren jahrzehntelangen Einsatz ist. Darüber hinaus verweist er darauf, dass das IP-Masquerading beziehungsweise NAT nicht nur das Problem behebt, dass es zu wenig Internetadressen gibt, sondern auch dazu führt, dass die hinter der Firewall liegenden Rechner von außen nicht mehr angreifbar sind. Er prophezeit ein Anwachsen der Urheberrechtsverletzungen. Da unter IPv6 jeder Rechner seine eigene IP-Adresse hat, würde dann der Aufbau von Tauschnetzen, wie sie erstmals von Napster eingeführt wurden, wesentlich effizienter laufen. Diese Aussage muss man wohl als Polemik betrachten. Im Gegensatz zu seiner Sorge um den Durchsatz von IPv6 führt er hier die bessere Funktionalität von IPv6 als Argumentationsgrundlage an. Mit einer etwas anders gelagerten Argumentation könnte man sogar IPv6 als »Napster-Killer« bezeichnen. Da

22 vgl. Nemeth/Snyder/Seebass/Hein: UNIX-Systemverwaltung. Markt+Technik – Prentice Hall, München, 2001. S. 357f.

23 vgl. Heise-Online-Meldung vom 14.1.2004 »Licht und Schatten bei IPv6«.

nun jeder beteiligte Rechner mit seiner Original-IP-Adresse angeschlossen ist, wird es für die Musik- und Filmindustrie sehr viel einfacher, die beteiligten Rechner zu identifizieren. Auch wenn man die Argumentation von Garfinkel also mit Vorsicht genießen muss, wird deutlich, dass es diverse Bedenken gegen die Einführung von IPv6 insbesondere in den USA gibt.

Zusammenfassend gibt es inzwischen im europäischen und amerikanischen Raum keinen zwingenden Grund für einen Umstieg auf IPv6. Sollte es gelingen, die bisherigen Ressourcen mit den Asiaten zu teilen, könnte das sogar dazu führen, dass es bei IPv4 bleibt. Andernfalls kann man davon ausgehen, dass IPv6 in Asien irgendwann zum Standard wird. Inwieweit die USA diesen Umstieg mitmachen, ist offen. Allerdings sind die USA verpflichtet ab 2008 auf den großen Backbones mit IPv6 zu arbeiten. Der Wechsel zu IPv6 wird sicher nicht problemlos ablaufen. In vieler Hinsicht ähnelt das Szenario dem des Jahr-2000-Problems. Man kann nicht sicher sagen, welche Systeme wirklich davon betroffen sind und wie viele Ausfälle durch den Wechsel entstehen werden. Auf der anderen Seite ist auch beim Jahr 2000 der Untergang der zivilisierten Welt vorhergesagt worden und doch nicht eingetreten.[24] Der Hauptunterschied zwischen beiden Situationen ist aber, dass das Jahr 2000 irgendwann einmal kommen musste, für alle Systeme zeitgleich, und das ausgerechnet an Silvester.

24 Zumindest sind die diesbezüglichen Rückschritte der Zivilisation nicht den Computern anzulasten.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z