Kapitel 5. Netzwerkkonfiguration

Inhaltsverzeichnis

Tipp

Bezüglich einer grundsätzlichen Anleitung zum Thema Netzwerk unter GNU/Linux lesen Sie den Linux Network Administrators Guide.

Tipp

For modern Debian specific guide to the networking, read The Debian Administrator's Handbook — Configuring the Network.

Warnung

This chapter is getting outdated since this is based on Debian 6.0 (**Squeeze**) released in 2011.

Tipp

Obwohl dieses Dokument immer noch das alte ifconfig(8) mit IPv4 für die Netzwerkkonfigurations-Beispiele verwendet, wechselte Debian mit der **Wheezy**-Veröffentlichung zu ip(8) mit IPv4 + IPv6. Änderungsvorschläge, um dieses Dokument auf aktuellen Stand zu bringen, sind gerne gesehen.

Tipp

Unter systemd kann networkd für die Netzwerkverwaltung genutzt werden; lesen Sie dazu systemd-networkd(8).

5.1. Die elementare Netzwerkinfrastruktur

Lassen Sie uns einen Blick auf die elementare Netzwerkinfrastruktur eines modernen Debian-Systems werfen:

Tabelle 5.1. Liste von Werkzeugen zur Netzwerkkonfiguration

Pakete	Popcon	Größe	Art	Beschreibung
ifupdown	V:582, I:995	184	config::ifupdown	standardisiertes Werkzeug zum Aktivieren und Deaktivieren des Netzwerks (Debian-spezifisch)
ifplugd	V:3, I:15	352	"	kabelgebundenes Netzwerk automatisch verwalten
ifupdown-extra	V:0, I:1	92	"	Netzwerk-Testskript zur Aufwertung des "**ifupdown**"-Pakets
ifmetric	V:0, I:1	21	"	Routenmetrik für eine Netzwerkschnittstelle setzen
guessnet	V:0, I:1	402	"	Mapping-Skript zur Aufwertung des "**ifupdown**"-Pakets mittels der "**/etc/network/interfaces**"-Datei
ifscheme	V:0, I:0	132	"	Mapping-Skript zur Aufwertung des "**ifupdown**"-Pakets
ifupdown-scripts-zg2	V:0, I:0	147	"	Zugschluss' Schnittstellenskripte für die manuelle ifupdown-Methode
network-manager	V:368, I:482	14208	config::NM	NetworkManager (Daemon): das Netzwerk automatisch verwalten
network-manager-gnome	V:252, I:426	6531	"	NetworkManager (GNOME-Frontend)
wicd	I:33	34	config::wicd	Manager für kabelgebundenes und kabelloses Netzwerk (Metapaket)
wicd-cli	V:0, I:2	58	"	Manager für kabelgebundenes und kabelloses Netzwerk (Befehlszeilen-Programm)
wicd-curses	V:1, I:5	174	"	Manager für kabelgebundenes und kabelloses Netzwerk (Curses-basiertes Programm)
wicd-daemon	V:28, I:38	951	"	Manager für kabelgebundenes und kabelloses Netzwerk (Daemon)

Pakete	Popcon	Größe	Art	Beschreibung
wicd-gtk	V:22, I:34	573	"	Manager für kabelgebundenes und kabelloses Netzwerk (GTK+-Programm)
iptables	V:246, I:996	1558	config::Netfilter	Administrationswerkzeuge für Paketfilterung und NAT (Netfilter)
iproute	V:200, I:637	22	config::iproute2	iproute2, IPv6 und andere erweiterte Netzwerkkonfiguration: ip(8), tc(8) usw.
ifrename	V:1, I:2	192	"	Netzwerkschnittstellen basierend auf verschiedenen statischen Kriterien umbenennen: ifrename(8)
ethtool	V:94, I:256	306	"	Eigenschaften von Ethernet-Geräten anzeigen oder ändern
iputils-ping	V:268, I:996	106	test::iproute2	Erreichbarkeit eines fernen Rechners über das Netzwerk testen, entweder mittels Rechnername oder IP-Addresse (iproute2)
iputils-arping	V:65, I:356	57	"	Erreichbarkeit eines fernen Rechners über das Netzwerk mittels seiner ARP-Addresse testen
iputils-tracepath	V:18, I:293	90	"	Netzwerkpfad zu einem fernen Rechner verfolgen
net-tools	V:465, I:998	883	config::net-tools	NET-3 Netzwerkprogramm-Sammlung (net-tools, IPv4-Netzwerkkonfiguration): ifconfig(8) usw.
inetutils-ping	V:0, I:1	329	test::net-tools	Erreichbarkeit eines fernen Rechners über das Netzwerk testen, entweder mittels Rechnername oder IP-Addresse (altes System, GNU)
arping	V:2, I:27	103	"	Erreichbarkeit eines fernen Rechners über das Netzwerk mittels seiner ARP-Addresse testen (altes System)
traceroute	V:77, I:993	154	"	Netzwerkpfad zu einem fernen Rechner verfolgen (altes System, Konsolenprogramm)
isc-dhcp-client	V:510, I:951	632	config::low-level	DHCP-Client
wpasupplicant	V:399, I:553	2396	"	clientseitige Unterstützung für WPA und WPA2 (IEEE 802.11i)
wpagui	V:0, I:4	808	"	Qt-GUI-Programm für wpa_supplicant
wireless-tools	V:61, I:278	325	"	Werkzeuge zum Bearbeiten der Linux Wireless Extensions
ppp	V:169, I:529	927	"	PPP-/PPPoE-Verbindung mit **chat**
pppoeconf	V:0, I:13	290	config::helper	Konfigurations-Hilfswerkzeug für PPPoE-Verbindungen
pppconfig	V:1, I:4	782	"	Konfigurations-Hilfswerkzeug für PPP-Verbindungen mit **chat**
wvdial	V:1, I:9	276	"	Konfigurations-Hilfswerkzeug für PPP-Verbindungen mit **wvdial** und **ppp**
mtr-tiny	V:8, I:63	105	test::low-level	Netzwerkpfad zu einem fernen Rechner verfolgen (Curses-basiert)
mtr	V:6, I:39	150	"	Netzwerkpfad zu einem fernen Rechner verfolgen (Curses- und GTK+-basiert)
gnome-nettool	V:16, I:295	2111	"	Werkzeuge für allgemeine Netzwerkinformations-Operationen (GNOME)
nmap	V:52, I:457	21032	"	Netzwerk-Mapper/Port-Scanner (Nmap, konsolen-basiert)

Pakete	Popcon	Größe	Art	Beschreibung
zenmap	V:3, I:13	2729	"	Netzwerk-Mapper/Port-Scanner (GTK+-basiert)
tcpdump	V:23, I:194	1132	"	Netzwerkverkehr-Analysator (Tcpdump, konsolen-basiert)
wireshark	V:5, I:66	91	"	Netzwerkverkehr-Analysator (Wireshark, GTK+-basiert)
tshark	V:3, I:36	361	"	Netzwerkverkehr-Analysator (konsolen-basiert)
nagios3	I:9	9	"	Überwachungs- und Verwaltungssystem für Rechner, Dienste und Netzwerke (Nagios)
tcptrace	V:0, I:2	389	"	eine Zusammenfassung von Verbindungen auf Basis der **tcpdump**-Ausgabe erstellen
snort	V:1, I:2	1920	"	flexibles Einbruchmeldesystem für das Netzwerk (Snort)
ntop	V:4, I:9	1617	"	Daten über die Netzwerknutzung im Webbrowser anzeigen
dnsutils	V:88, I:909	344	"	Netzwerk-Clients, die mit BIND bereitgestellt werden: nslookup(8), nsupdate(8), dig(8)
dlint	V:1, I:28	96	"	DNS-Zoneninformationen mittels Nameserver-Abfragen überprüfen
dnstracer	V:0, I:2	81	"	eine Verkettung von DNS-Servern zu ihrer Quelle verfolgen

5.1.1. Die Auflösung des Rechnernamens

Die Auflösung des Rechnernamens (hostname) wird derzeit auch durch den NSS-(Name-Service-Switch-)Mechanismus unterstützt. Die Auflösung läuft wie folgt ab:

1. Die "**/etc/nsswitch.conf**"-Datei mit Einträgen wie "**hosts: files dns**" bestimmt die Reihenfolge der Rechnernamenauflösung. (Dies ersetzt die alte Funktionalität der "**order**"-Einträge in "**/etc/host.conf**".)

2. Als erstes wird in diesem Beispiel die **files**-Methode aufgerufen. Wenn der Rechnername in der "**/etc/hosts**"-Datei gefunden wird, werden alle gültigen Adressen für den Rechner ausgegeben und die Abfrage wird beendet. (Die "**/etc/host.conf**"-Datei enthält "**multi on**".)

3. Dann wird die **dns**-Methode wird aufgerufen. Wenn der Rechnername über das Internet Domain Name System (DNS) (definiert über die Datei "**/etc/resolv.conf**") gefunden wird, werden alle dafür gültigen Adressen ausgegeben und die Abfrage wird beendet.

Die "**/etc/hosts**"-Datei sieht zum Beispiel so aus:

```
127.0.0.1 localhost
127.0.1.1 <rechnername>

# The following lines are desirable for IPv6 capable hosts
# (die folgenden Zeilen sind für IPv6-fähige Rechner wünschenswert).
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Jede Zeile beginnt mit einer IP-Addresse und dahinter steht jeweils der zugeordnete Rechnername.

Die IP-Adresse **127.0.1.1** in der zweiten Zeile dieses Beispiels ist auf einigen anderen Unix-ähnlichen Systemen möglicherweise nicht vorhanden. Der Debian Installer erstellt diesen Eintrag für Systeme ohne feste IP-Adresse als provisorische Lösung für einige Software-Produkte (z.B. GNOME), wie in Fehler #719621 dokumentiert.

Der Eintrag <rechnername> entspricht dem in"**/etc/hostname**" festgelegten Rechnernamen.

Auf Systemen mit einer festen IP-Adresse sollte allerdings diese feste IP-Adresse statt der **127.0.1.1** verwendet werden.

Bei Systemen mit einer festen IP-Adresse und einem voll qualifizierten Domain-Namen (FQDN), bereitgestellt durch das Domain Name System (DNS), sollte <rechnername>.<domain-name> verwendet werden statt nur <rechnername>.

Die Datei "**/etc/resolv.conf**" ist eine statische Datei, falls das Paket **resolvconf** nicht installiert ist. Falls das Paket installiert ist, ist dies ein symbolischer Link. In beiden Fällen enthält es Informationen zur Initialisierung der Namensauflösungs-Routinen. Wenn das DNS zum Beispiel über die IP "**192.168.11.1**" erreichbar ist, enthält sie Folgendes:

```
nameserver 192.168.11.1
```

Das **resolvconf**-Paket macht "**/etc/resolv.conf**" zu einem symbolischen Link und verwaltet ihren Inhalt automatisch über die Hook-Skripte.

Bei PC-Arbeitsplatz-Rechnern in einer typischen LAN-Umgebung kann der Rechnername zusätzlich zu den grundlegenden **files**- und **dns**-Methoden auch über Multicast DNS (mDNS, Zeroconf) aufgelöst werden:

- Avahi stellt ein Rahmenwerk für Multicast-DNS-Diensteabfragen auf Debian-Systemen bereit.
- Es ist ein Äquivalent zu Apple Bonjour / Apple Rendezvous.
- Das **libnss-mdns**-Plugin-Paket bietet Rechnernamensauflösung via mDNS für die GNU Name-Service-Switch-(NSS-)Funktionalität der GNU C-Bibliothek (glibc).
- Die Datei "**/etc/nsswitch.conf**" sollte Einträge wie "**hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4**" enthalten.
- Rechnernamen, die mit der Pseudo-Top-Level Domain (TLD) ".local" enden, werden aufgelöst.
- Die mDNS IPv4 link-lokale Multicast-Adresse "**224.0.0.251**" oder ihr IPv6-Äquivalent "**FF02::FB**" wird verwendet, um DNS-Abfragen für einen auf "**.local**" endenden Namen durchzuführen.

Rechnernamensauflösung über das veraltete NETBios over TCP/IP, das von älteren Windows-Systemen verwendet wurde, kann über die Installation des Pakets **winbind** realisiert werden. Die "**/etc/nsswitch.conf**"-Datei sollte Einträge wie "**hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4 wins**" enthalten, um diese Funktionalität zu aktivieren. (Moderne Windows-Systeme verwenden normalerweise die **dns**-Methode zur Rechnernamensauflösung.)

Anmerkung

Die Ausweitung generischer Top-Level-Domains (gTLD) im Domain-Name-System ist in Arbeit. Achten Sie bei Auswahl von Domain-Namen, die nur im lokalen Netzwerk verwendet werden sollen, auf Namenskollisionen.

5.1.2. Der Netzwerkschnittstellenname

Der Netzwerkschnittstellenname, z.B. **eth0**, wird jeglicher Hardware im Linux-Kernel bei der Erkennung durch den User-Space-Konfigurationsmechanismus **udev** zugewiesen (lesen Sie dazu Abschnitt 3.3, „Das udev-System“). Der Netzwerkschnittstellenname wird in ifup(8) und interfaces(5) als **physical interface** (physikalische Schnittstelle) bezeichnet.

Um über MAC-Adresse (und anderes) sicherzustellen, dass jede Netzwerkschnittstelle bei jedem Neustart dauerhaft die gleiche Bezeichnung hat, gibt es eine Regeldatei "**/etc/udev/rules.d/70-persistent-net.rules**". Diese Datei wird automatisch durch das "**/lib/udev/write_net_rules**"-Programm generiert, unter Umständen ausgeführt von der "**persistent-net-generator.rules**"-Regeldatei. Sie können sie editieren, um die Namensregeln zu ändern.

Achtung

Wenn Sie die "**/etc/udev/rules.d/70-persistent-net.rules**"-Regeldatei verändern, müssen Sie für jede Regel eine eigene Zeile verwenden und die Buchstaben in der MAC-Adresse müssen Kleinbuchstaben sein. Wenn Sie zum Beispiel Einträge für "FireWire device" und "PCI device" in dieser Datei finden, möchten Sie vielleicht das Gerät unter "PCI device" mit **eth0** benennen und es als primäre Netzwerkschnittstelle konfigurieren.

5.1.3. Der Netzwerkadressbereich für das LAN

Wir wollen uns an die IPv4 32-Bit-Adressbereiche erinnern, die durch die rfc1918 für jede Klasse zur Verwendung in Local Area Networks (LANs) reserviert sind. Diese Adressen werden bestimmt nicht mit irgendwelchen Adressen im Internet kollidieren.

Tabelle 5.2. Liste der Netzwerkadressbereiche

Klasse	Netzwerkadressen	Netzmaske	Netzmaske /Bits	Anzahl der Unternetzwerke
A	10.x.x.x	255.0.0.0	/8	1
B	172.16.x.x — 172.31.x.x	255.255.0.0	/16	16
C	192.168.0.x — 192.168.255.x	255.255.255.0	/24	256

Anmerkung

Wenn eine dieser Adressen einem Rechner zugewiesen ist, kann dieser Rechner das Internet nicht direkt erreichen, sondern muss ein Gateway verwenden, der als Proxy für verschiedene Dienste dient, oder er nutzt Network Address Translation (NAT). Ein Breitband-Router nutzt üblicherweise NAT für das Anwender-Netzwerk.

5.1.4. Unterstützung für Netzwerkgeräte

Der größte Teil verfügbarer Netzwerk-Hardware wird durch das Debian-System unterstützt; es gibt einige Geräte, die laut DFSG nicht-freie Firmware für den Betrieb erfordern. Lesen Sie dazu Abschnitt 9.9.6, „Hardware-Treiber und Firmware“.

5.2. Die moderne Netzwerkkonfiguration für Arbeitsplatzsysteme

Debian-Systeme der Version 6.0 (**Squeeze**) und später können Netzwerkverbindungen über Software-Daemons wie NetworkManager (NM) (network-manager und zugehörige Pakete) oder Wicd (wicd und zugehörige Pakete) verwalten.

- Sie haben ihre eigenen grafischen GUI- und Befehlszeilen-Programme als Bedienoberfläche.
- Sie haben ihre eigenen Daemons als Unterbau.
- Sie erlauben eine einfache Verbindung Ihres Systems mit dem Internet.
- Sie ermöglichen eine problemlose Verwaltung von kabelgebundenen und kabellosen Netzwerkkonfigurationen.
- Sie erlauben uns, das Netzwerk unabhängig vom althergebrachten **ifupdown** zu konfigurieren.

Anmerkung

Verwenden Sie diese automatischen Netzwerkkonfigurations-Werkzeuge nicht für Server. Sie sind primär für die Nutzung auf Arbeitsplatzrechnern oder Laptops gedacht.

Diese modernen Werkzeuge müssen korrekt konfiguriert werden, um Konflikte mit dem **ifupdown**-Paket und seiner Konfigurationsdatei "**/etc/network/interfaces**" zu vermeiden.

Tipp

Under systemd, the network may be configured in **/etc/systemd/network/** instead. See systemd-resolved(8), resolved.conf(5), and systemd-networkd(8).

Anmerkung

Einige Funktionalitäten dieser automatischen Netzwerkkonfigurations-Werkzeuge könnten unter Rückentwicklungen leiden. Sie sind nicht so robust wie das altbewährte **ifupdown**. Überprüfen Sie die Fehlerberichte zu network-manager und wicd bezüglich aktueller Probleme oder Einschränkungen.

5.2.1. Grafische Netzwerkkonfigurations-Werkzeuge

Offizielle Dokumentation für NM und Wicd unter Debian ist in "**/usr/share/doc/network-manager/README.Debian**" respektive "**/usr/share/doc/wicd/README.Debian**" verfügbar.

Grundsätzlich läuft die Netzwerkkonfiguration für Arbeitsplatzsysteme wie folgt ab:

1. Fügen Sie den Benutzer, der sich am Arbeitsplatz anmeldet, z.B. **foo**, mit folgendem Befehl zur Gruppe "**netdev**" hinzu (alternativ kann dies in modernen Arbeitsplatz-Umgebungen wie GNOME oder KDE auch automatisch über D-bus erledigt werden):

```
$ sudo adduser foo netdev
```

2. Halten Sie die Konfiguration in "**/etc/network/interfaces**" so einfach wie hier:

```
auto lo
iface lo inet loopback
```

3. Starten Sie NM bzw. Wicd mit einem der folgenden Befehle neu:

```
$ sudo /etc/init.d/network-manager restart
```

```
$ sudo /etc/init.d/wicd restart
```

4. Konfigurieren Sie Ihr Netzwerk über die grafische GUI-Oberfläche.

Anmerkung

Um Konflikte mit **ifupdown** zu vermeiden, werden nur Schnittstellen, die **nicht** in "**/etc/network/interfaces**" aufgelistet sind, von NM oder Wicd verwaltet.

Tipp

Wenn Sie die Fähigkeiten von NM erweitern möchten, suchen Sie nach entsprechenden Plugin-Modulen und zusätzlichen Paketen wie **network-manager-openconnect**, **network-manager-openvpn-gnome**, **network-manager-pptp-gnome**, **mobile-broadband-provider-info**, **gnome-bluetooth** usw. Das gleiche gilt für Wicd.

Achtung

Diese automatischen Netzwerkkonfigurations-Werkzeuge sind unter Umständen nicht kompatibel mit esoterischen Konfigurationen des althergebrachten **ifupdown** in "**/etc/network/interfaces**", wie denen in Abschnitt 5.5, „Die grundlegende Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)“ und Abschnitt 5.6, „Die erweiterte Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)“. Überprüfen Sie die Fehlerberichte zu network-manager und wicd bezüglich aktueller Probleme oder Einschränkungen.

5.3. Die Netzwerkverbindung und -konfiguration nach altbewährtem System

Wenn die Methode aus Abschnitt 5.2, „Die moderne Netzwerkkonfiguration für Arbeitsplatzsysteme“ Ihren Ansprüchen nicht genügt, sollten Sie die altbewährte Methode für Netzwerkverbindung und -konfiguration verwenden, die erheblich einfachere Werkzeuge nutzt.

Die altbewährte Netzwerkverbindung ist für jede einzelne Methode unterschiedlich (Näheres in Abschnitt 5.4, „Die Netzwerkverbindungs-Methoden (nach altbewährtem System)“).

Es gibt unter Linux zwei Arten von Programmen für die Arbeiten im Untergrund der Netzwerkkonfiguration (lesen Sie dazu Abschnitt 5.7.1, „iproute2-Befehle“):

- Die alten net-tools-Programme (ifconfig(8), …) stammen vom Linux NET-3 Netzwerk-System. Die meisten davon sind mittlerweile überholt.
- Neue Linux iproute2-Programme (ip(8), …) bilden das aktuelle Linux-Netzwerk-System.

Obwohl diese Basis-Netzwerk-Programme sehr leistungsfähig sind, ist Ihre Anwendung etwas umständlich. Daher wurden übergeordnete Konfigurationssysteme entwickelt.

Das **ifupdown**-Paket ist der De-Facto-Standard für diese übergeordneten Netzwerkkonfigurationssysteme unter Debian. Es ermöglicht Ihnen die Aktivierung des Netzwerks z.B. über ein einfaches "**ifup eth0**". Seine Konfigurationsdatei ist "**/etc/network/interfaces**" und deren typischen Inhalte sind Folgende:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Das **resolvconf**-Paket wurde als Ergänzung zum **ifupdown**-System erstellt, um eine problemlose Neukonfiguration der Netzwerkadressauflösung durch automatisches Neuerstellen der Konfigurationsdatei "**/etc/resolv.conf**" zu ermöglichen. Die meisten Netzwerkkonfigurationspakete in Debian wurden entsprechend angepasst, so dass Sie jetzt das **resolvconf**-Paket verwenden (lesen Sie auch "**/usr/share/doc/resolvconf/README.Debian**").

Hilfsskripte zum **ifupdown**-Paket wie **ifplugd**, **guessnet**, **ifscheme** usw. wurden erstellt, um die dynamische Konfiguration von Netzwerkumgebungen (wie z.B. die eines mobilen PC an kabelgebundenen LAN) zu automatisieren. Sie sind zwar relativ schwierig zu benutzen, funktionieren aber gut mit dem vorhandenen **ifupdown**-System.

Diese Hilfsskripte werden im Detail und mit Beispielen in Abschnitt 5.5, „Die grundlegende Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)“ und Abschnitt 5.6, „Die erweiterte Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)“) beschrieben.

5.4. Die Netzwerkverbindungs-Methoden (nach altbewährtem System)

Achtung

Die Methoden zum Testen der Netzwerkverbindung in diesem Abschnitt sind für Testzwecke gedacht. Sie sollten sie nicht direkt für die täglichen

Netzwerkverbindungen nutzen. Es wird empfohlen, stattdessen NM, Wicd oder das **ifupdown**-Paket zu verwenden (Näheres in Abschnitt 5.2, „Die moderne Netzwerkkonfiguration für Arbeitsplatzsysteme“ und Abschnitt 5.5, „Die grundlegende Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)“).

Typische Netzwerkverbindungsmethoden und Verbindungspfade für einen PC können wie folgt zusammengefasst werden:

Tabelle 5.3. Liste von Netzwerkverbindungsmethoden und Verbindungspfaden

PC	Verbindungsmethoden	Verbindungspfad
Serieller Port (**ppp0**)	PPP	⇔ Modem ⇔ Telefonverbindung ⇔ Einwahl-Zugangspunkt ⇔ Internet-Anbieter
Ethernet-Port (**eth0**)	PPPoE/DHCP/Statisch	⇔ Breitband-Modem ⇔ Breitband-Dienst ⇔ Breitband-Zugangspunkt ⇔ Internet-Anbieter
Ethernet-Port (**eth0**)	DHCP/Statisch	⇔ LAN ⇔ Breitband-Router mit Network Address Translation (NAT) (⇔ Breitband-Modem …)

Hier eine Zusammenfassung der Konfigurationsskripte für die verschiedenen Verbindungsmethoden:

Tabelle 5.4. Liste der Netzwerkverbindungskonfigurationen

Verbindungsmethoden	Konfiguration	Backend-(Unterbau-)Paket
PPP	**pppconfig** zur Erzeugung von deterministischer Chat-Verbindung	**pppconfig**, **ppp**
PPP (alternativ)	**wvdialconf** zur Erzeugung von heuristischer Chat-Verbindung	**ppp**, **wvdial**
PPPoE	**pppoeconf** zur Erzeugung von deterministischer Chat-Verbindung	**pppoeconf**, **ppp**
DHCP	definiert in "**/etc/dhcp/dhclient.conf**"	**isc-dhcp-client**
statische IP (IPv4)	definiert in "**/etc/network/interfaces**"	**iproute2** oder **net-tools** (veraltet)
statische IP (IPv6)	definiert in "**/etc/network/interfaces**"	**iproute2**

Die Abkürzungen bei Netzwerkverbindungen haben folgende Bedeutungen:

Tabelle 5.5. Liste der Abkürzungen bei Netzwerkverbindungen

Abkürzung	Bedeutung
POTS	plain old telephone service (alter reiner Telefondienst)
BB	Breitband
BB-service	z.B. Digital Subscriber Line (DSL), TV-Kabel oder Fiber To The Premises (FTTP, Glasfaser)
BB-modem	z.B. DSL-Modem, Kabel-Modem oder Optical Network Terminal (ONT) für Glasfaser
LAN	Local Area Network (lokales Netzwerk)
WAN	Wide Area Network (großräumiges Netzwerk, z.B. das Internet)
DHCP	Dynamic Host Configuration Protocol
PPP	Point-to-Point Protocol
PPPoE	Point-to-Point Protocol over Ethernet
ISP	Internet Service Provider (Internet-Diensteanbieter)

Anmerkung

WAN-Verbindungsdienste über TV-Kabel werden grundsätzlich mit DHCP oder PPPoE ausgeliefert, diejenigen über ADSL und FTTP mit PPPoE. Sie müssen Ihren ISP kontaktieren, um die exakten Konfigurationsanforderungen der WAN-Verbindung zu erfragen.

Anmerkung

Wenn ein Breitband-Router verwendet wird, um eine Heim-Netzwerkumgebung aufzubauen, werden PCs im LAN mittels Network Address Translation (NAT) durch den Breitband-Router mit dem WAN (Internet) verbunden. In solchen Fällen haben die Netzwerkschnittstellen der PCs im LAN entweder eine statische IP-Adresse oder werden via DHCP vom Breitband-Router bedient. Die Internet-Verbindung des Breitband-Routers muss entsprechend den Anweisungen des ISP konfiguriert werden.

5.4.1. Die DHCP-Verbindung über Ethernet

Das typische moderne Netzwerk für Heim und kleine Unternehmen (LAN) wird über einen Breitband-Router mit dem Internet (WAN) verbunden. Das LAN hinter dem Router wird üblicherweise über einen Dynamic Host Configuration Protocol (DHCP)-Server, der auf dem Router läuft, verwaltet.

Installieren Sie einfach das Paket **`isc-dhcp-client`**, um das durch DHCP verwaltete Ethernet zu nutzen.

Lesen Sie dazu auch dhclient.conf(5).

5.4.2. Statische IP-Verbindung über Ethernet

Es gibt keine speziellen Anforderungen, um das Ethernet mit statischer IP zu nutzen.

5.4.3. Die PPP-Verbindung mit pppconfig

Das Konfigurationsskript **`pppconfig`** konfiguriert die PPP-Verbindung interaktiv, indem einfach die folgenden Daten abgefragt werden:

- Telefonnummer;
- Name des ISP;
- Passwort vom ISP;
- Anschlußgeschwindigkeit;
- Modemanschluß für die Kommunikation;
- Authentifizierungsmethode.

Tabelle 5.6. Liste der Konfigurationsdateien für eine PPP-Verbindung mit pppconfig

Datei	Funktion
`/etc/ppp/peers/<isp-name>`	Durch **`pppconfig`** erzeugte Konfigurationsdatei für **`pppd`**, spezifisch für den ISP <isp-name>
`/etc/chatscripts/<isp-name>`	Durch **`pppconfig`** erzeugte Konfigurationsdatei für **`chat`**, spezifisch für den ISP <isp-name>
`/etc/ppp/options`	Grundlegende Ausführungsparameter für **`pppd`**
`/etc/ppp/pap-secret`	Authentifizierungsdaten für PAP (unsicher)
`/etc/ppp/chap-secret`	Authentifizierungsdaten für CHAP (höhere Sicherheit)

Achtung

Werden die Befehle **pon** und **poff** ohne weitere Argumente aufgerufen, wird automatisch "**provider**" als "<isp-name>" genutzt.

Sie können die Konfiguration testen, indem Sie die zugrundeliegenden Konfigurationswerkzeuge wie folgt aufrufen:

```
$ sudo pon <isp-name>
...
$ sudo poff <isp-name>
```

Lesen Sie dazu auch "**/usr/share/doc/ppp/README.Debian.gz**".

5.4.4. Die alternative PPP-Verbindung mit wvdialconf

Ein anderer Ansatz, pppd(8) zu verwenden, ist der Aufruf über wvdial(1) aus dem gleichnamigen Paket. Statt dass **pppd** den chat(8)-Prozess zwecks Einwahl und Aushandlung der Verbindung aufruft, übernimmt **wvdial** die Einwahl und den anfänglichen Verbindungsaufbau selbst und startet dann **pppd** für den Rest.

Das Konfigurationsskript **wvdialconf** konfiguriert die PPP-Verbindung interaktiv, indem folgende Daten abgefragt werden:

- Telefonnummer;
- Name des ISP;
- Passwort vom ISP;

wvdial stellt in den meisten Fällen erfolgreich die Verbindung her und verwaltet automatisch eine Liste mit den Authentifizierungsdaten.

Tabelle 5.7. Liste der Konfigurationsdateien für eine PPP-Verbindung mit wvdialconf

Datei	Funktion
/etc/ppp/peers/wvdial	Durch **wvdialconf** erzeugte Konfigurationsdatei für **pppd**, spezifisch für **wvdial**
/etc/wvdial.conf	Durch **wvdialconf** erzeugte Konfigurationsdatei
/etc/ppp/options	Grundlegende Ausführungsparameter für **pppd**
/etc/ppp/pap-secret	Authentifizierungsdaten für PAP (unsicher)
/etc/ppp/chap-secret	Authentifizierungsdaten für CHAP (höhere Sicherheit)

Sie können die Konfiguration testen, indem Sie die zugrundeliegenden Konfigurationswerkzeuge wie folgt aufrufen:

```
$ sudo wvdial
...
$ sudo killall wvdial
```

Lesen Sie dazu auch wvdial(1) und wvdial.conf(5).

5.4.5. Die PPPoE-Verbindung mit pppoeconf

Wenn Ihr ISP Ihnen eine PPPoE-Verbindung anbietet und Sie sich entscheiden, den PC direkt mit dem WAN (Internet) zu verbinden, muss das Netzwerk Ihres PCs mit PPPoE (PPP over Ethernet) konfiguriert werden. Das Konfigurationsskript **pppoeconf** konfiguriert die PPPoE-Verbindung durch interaktive Abfragen.

Die Konfigurationsdateien sind folgende:

Tabelle 5.8. Liste der Konfigurationsdateien für eine PPPoE-Verbindung mit pppoeconf

Datei	Funktion
/etc/ppp/peers /dsl-provider	Durch **pppoeconf** erzeugte Konfigurationsdatei für **pppd**, spezifisch für **pppoe**
/etc/ppp/options	Grundlegende Ausführungsparameter für **pppd**
/etc/ppp/pap-secret	Authentifizierungsdaten für PAP (unsicher)
/etc/ppp/chap-secret	Authentifizierungsdaten für CHAP (höhere Sicherheit)

Sie können die Konfiguration testen, indem Sie die zugrundeliegenden Konfigurationswerkzeuge wie folgt aufrufen:

```
$ sudo /sbin/ifconfig eth0 up
$ sudo pon dsl-provider
...
$ sudo poff dsl-provider
$ sudo /sbin/ifconfig eth0 down
```

Lesen Sie dazu auch "**/usr/share/doc/pppoeconf/README.Debian**".

5.5. Die grundlegende Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)

Zur Einrichtung eines traditionellen TCP/IP-Netzwerks wird auf einem Debian-System das **ifupdown**-Paket als hochentwickeltes Werkzeug verwendet. Es gibt zwei typische Fälle:

- Auf Systemen mit einer **dynamischen IP-Adresse** wie mobilen PCs sollten Sie das TCP/IP-Netzwerk **mit** dem **resolvconf**-Paket einrichten; dies ermöglicht Ihnen, Ihre Netzwerkkonfiguration einfach auf eine andere Situation umzuschalten (Näheres in Abschnitt 5.5.4, „Die über DHCP bediente Netzwerkschnittstelle“).
- Auf Systemen mit einer **statischen IP-Adresse** wie z.B. auf Servern sollten Sie das TCP/IP-Netzwerk **ohne** das **resolvconf**-Paket einrichten und so das System einfach halten (Weiteres finden Sie in Abschnitt 5.5.5, „Die Netzwerkschnittstelle mit statischer IP-Adresse“).

Diese traditionellen Methoden der Einrichtung sind sehr nützlich, wenn Sie erweiterte Konfigurationen benötigen; hier folgen weitere Details.

Das **ifupdown**-Paket stellt ein standardisiertes Rahmenwerk für hochentwickelte Netzwerkkonfiguration im Debian-System bereit. In diesem Abschnitt lernen wir die grundlegende Netzwerkkonfiguration mittels **ifupdown** mit vereinfachten Erklärungen und vielen typischen Beispielen kennen.

5.5.1. Die Befehlssyntax in vereinfachter Form

Das **ifupdown**-Paket enthält zwei Befehle: ifup(8) and ifdown(8). Sie bieten hochentwickelte Netzwerkkonfiguration, gesteuert durch die Konfigurationsdatei "/etc/network/interfaces".

Tabelle 5.9. Liste der Befehle zur grundlegenden Netzwerkkonfiguration mit ifupdown

Befehl	Aktion
ifup eth0	die Netzwerkschnittstelle **eth0** mit der Konfiguration für **eth0** aktivieren, falls ein Eintrag für "**iface eth0**" existiert.
ifdown eth0	die Netzwerkschnittstelle **eth0** mit der Konfiguration für **eth0** deaktivieren, falls ein Eintrag für "**iface eth0**" existiert.

Warnung

Verwenden Sie keine der im Untergrund arbeitenden Werkzeuge wie ifconfig(8) und ip(8), um Schnittstellen zu konfigurieren, die sich im Status **up** (aktiv) befinden.

Anmerkung

Es gibt keinen Befehl **ifupdown**.

5.5.2. Der grundlegende Aufbau von "/etc/network/interfaces"

Der Aufbau von "**/etc/network/interfaces**" (auch in interfaces(5) beschrieben) kann wie folgt zusammengefasst werden:

Tabelle 5.10. Liste der Einträge in "/etc/network/interfaces"

Eintrag	Bedeutung
"**auto <name>**"	die Schnittstelle <name> während des Systemstarts aktivieren
"**allow-auto <name>**"	"
"**allow-hotplug <name>**"	die Schnittstelle <name> aktivieren, wenn der Kernel ein Hotplug-Ereignis von der Schnittstelle empfängt
Zeilen beginnend mit "**iface <konfig-name> …**"	Festlegung der Netzwerkkonfiguration <konfig-name>
Zeilen beginnend mit "**mapping <muster-für-name>**"	Festlegung des Wertes zum Mapping von <konfig_name> für eine auf <muster-für-name> passende Schnittstelle
eine mit "#" beginnende Zeile	wird als Kommentar ignoriert (Kommentare am Ende einer Zeile werden **nicht** unterstützt)
eine mit "\" endende Zeile	Konfiguration in der nächsten Zeile fortsetzen

Einträge beginnend mit **iface** haben folgenden Aufbau:

```
iface <konfig-name> <addressfamilie> <methodenname>
 <option1> <wert1>
 <option2> <wert2>
 ...
```

Bei Basiskonfigurationen werden **mapping**-Einträge nicht verwendet; es wird stattdessen der Name der Schnittstelle als Konfigurationsname genutzt (lesen Sie dazu Abschnitt 5.6.5, „Der mapping-Eintrag“).

Warnung

Geben Sie keine doppelten "**iface**"-Einträge für eine Netzwerkschnittstelle in "**/etc/network/interfaces**" an.

5.5.3. Die loopback-Netzwerkschnittstelle

Folgender Eintrag in der Datei "**/etc/network/interfaces**" aktiviert die loopback-Netzwerkschnittstelle **lo** während des Systemstarts (über den **auto**-Eintrag).

```
auto lo
iface lo inet loopback
```

Dieser Eintrag existiert immer in "**/etc/network/interfaces**".

5.5.4. Die über DHCP bediente Netzwerkschnittstelle

Nachdem das System gemäß Abschnitt 5.4.1, „Die DHCP-Verbindung über Ethernet“ vorbereitet wurde, wird eine über DHCP verwaltete Netzwerkschnittstelle mit einem Eintrag wie dem folgenden in "**/etc/network/interfaces**" konfiguriert:

```
allow-hotplug eth0
iface eth0 inet dhcp
```

Wenn der Linux-Kernel die physikalische Schnittstelle **eth0** detektiert, führt die Zeile **allow-hotplug** dazu, dass **ifup** automatisch die Schnittstelle aktiviert, und aufgrund des **iface**-Eintrags wird DHCP zur Konfiguration verwendet.

5.5.5. Die Netzwerkschnittstelle mit statischer IP-Adresse

Eine Netzwerkschnittstelle mit statischer (fester) IP-Adresse wird über einen Eintrag wie dem folgenden in "**/etc/network/interfaces**" konfiguriert:

```
allow-hotplug eth0
iface eth0 inet static
 address 192.168.11.100
 netmask 255.255.255.0
 gateway 192.168.11.1
 dns-domain example.com
 dns-nameservers 192.168.11.1
```

Detektiert der Linux-Kernel die physikalische Schnittstelle **eth0**, aktiviert **ifup** aufgrund der **allow-hotplug**-Zeile automatisch die Schnittstelle und nutzt gemäß dem **iface**-Eintrag die statische IP-Adresse für die Konfiguration.

Dabei gehe ich von folgenden Annahmen aus:

- IP-Adressbereich für das LAN-Netzwerk: **192.168.11.0** - **192.168.11.255**
- IP-Adresse des Gateways: **192.168.11.1**
- IP-Adresse des PCs: **192.168.11.100**
- Das **resolvconf**-Paket: installiert
- Domain-Name: "**example.com**"
- IP-Adresse des DNS-Servers: **192.168.11.1**

Falls das **resolvconf**-Paket nicht installiert ist, muss die Konfiguration des DNS manuell durchgeführt werden, indem "**/etc/resolv.conf**" wie folgt editiert wird:

```
nameserver 192.168.11.1
domain example.com
```

Achtung

Die in diesem Beispiel verwendeten IP-Adressen können nicht zwingend direkt kopiert werden. Sie müssen Adressen verwenden, die zu Ihrer jeweiligen Netzwerksituation passen.

5.5.6. Die Grundlagen von kabellosen LAN-Schnittstellen

Das Wireless LAN (kurz WLAN) bietet schnelle kabellose Verbindungen über eine Breitspektrum-Kommunikation in unlizensierten Funkfrequenzbändern, basierend auf den IEEE 802.11-Standards.

Die WLAN-Schnittstellen sind nahezu normale Ethernet-Schnittstellen, erfordern aber bei der Initialisierung einige Daten für Netzwerk-ID und Verschlüsselung. Die entsprechenden Netzwerkkonfigurations-Werkzeuge sind identisch mit denen für Ethernet-Schnittstellen, nur dass sich die Namen der Schnittstellen ein bisschen unterscheiden, sie heißen z.B. **eth1**, **wlan0**, **ath0**, **wifi0** ..., je nachdem, welche Kernel-Treiber zum Einsatz kommen.

Tipp

wmaster0 ist das Master-Gerät, das als internes Gerät nur von SoftMAC mit der neuen mac80211 API-of-Linux-Implementierung genutzt wird.

Hier einige für WLAN relevante Schlüsselbegriffe:

Tabelle 5.11. Liste der Abkürzungen für WLAN

Abkürzung	vollständiger Begriff	Bedeutung
NWID	Netzwerk-ID	16-Bit-Netzwerk-ID, die von alten (pre-802.11 / Vor-802.11-Standard-Implementierung) WaveLAN-Netzwerken genutzt wird (stark veraltet)
(E)SSID	(Extended) Service Set Identifier	Netzwerkname der Wireless Access Points (AP / drahtloser Zugangspunkt) zum Aufbau eines integrierten 802.11 Wireless-LAN, Domain-ID
WEP, (WEP2)	Wired Equivalent Privacy	64-Bit- (128-Bit-) Wireless-Verschlüsselungsstandard der ersten Generation mit 40-Bit-Schlüssel (veraltet)
WPA	Wi-Fi Protected Access	Wireless-Verschlüsselungsstandard der zweiten Generation (überwiegend 802.11i), kompatibel mit WEP
WPA2	Wi-Fi Protected Access 2	Wireless-Verschlüsselungsstandard der dritten Generation (vollständig 802.11i), nicht kompatibel mit WEP

Die Entscheidung, welches Protokoll letztlich verwendet wird, wird für gewöhnlich eingeschränkt durch die Fähigkeiten des verwendeten Wireless-Router.

5.5.7. Die Wireless-LAN-Schnittstelle mit WPA/WPA2

Zur Unterstützung des neuen WPA/WPA2 müssen Sie das **wpasupplicant**-Paket installieren.

Falls die IP-Adressen der WLAN-Verbindung per DHCP verwaltet werden, muss der Eintrag in der "**/etc/network/interfaces**"-Datei wie folgt aussehen:

```
allow-hotplug ath0
iface ath0 inet dhcp
 wpa-ssid homezone
 # der hexadezimale PSK ist mit einer Klartext-Passphrase verschlüsselt
 wpa-psk 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
```

Lesen Sie dazu auch "**/usr/share/doc/wpasupplicant/README.modes.gz**".

5.5.8. Die Wireless-LAN-Schnittstelle mit WEP

Zur Unterstützung des alten WEP müssen Sie das **wireless-tools**-Paket installieren. (Unter Umständen unterstützt Ihr Router nur diese unsichere Verschlüsselungsmethode, aber das ist immer noch besser als gar keine Verschlüsselung zu verwenden.)

> **Achtung**
>
> Bitte beachten Sie, dass der Netzwerkverkehr in einem WLAN mit WEP durch andere mitgeschrieben werden könnte.

Falls die IP-Adressen der WLAN-Verbindung per DHCP verwaltet werden, muss der Eintrag in der "**/etc/network/interfaces**"-Datei wie folgt aussehen:

```
allow-hotplug eth0
iface eth0 inet dhcp
 wireless-essid Home
 wireless-key1 0123-4567-89ab-cdef
 wireless-key2 12345678
 wireless-key3 s:password
 wireless-defaultkey 2
 wireless-keymode open
```

Lesen Sie dazu "**/usr/share/doc/wireless-tools/README.Debian**".

5.5.9. Die PPP-Verbindung

Sie müssen die PPP-Verbindung zunächst wie oben in Abschnitt 5.4.3, „Die PPP-Verbindung mit pppconfig“ beschrieben einrichten. Fügen Sie dann folgenden Eintrag für das primäre PPP-Gerät **ppp0** in die "**/etc/network/interfaces**"-Datei ein:

```
iface ppp0 inet ppp
 provider <isp-name>
```

5.5.10. Die alternative PPP-Verbindung

Sie müssen die alternative PPP-Verbindung mit **wvdial** zunächst wie in Abschnitt 5.4.4, „Die alternative PPP-Verbindung mit wvdialconf“) einrichten. Fügen Sie dann folgenden Eintrag für das primäre PPP-Gerät **ppp0** in die "**/etc/network/interfaces**"-Datei ein:

```
iface ppp0 inet wvdial
```

5.5.11. Die PPPoE-Verbindung

PCs, die über PPPoE direkt mit dem WAN (Internet) verbunden sind, müssen mittels PPPoE-Verbindung (wie in Abschnitt 5.4.5, „Die PPPoE-Verbindung mit pppoeconf“ beschrieben) eingerichtet werden. Fügen Sie dann folgenden Eintrag für das primäre PPPoE-Gerät **eth0** in die "**/etc/network/interfaces**"-Datei ein:

```
allow-hotplug eth0
iface eth0 inet manual
 pre-up /sbin/ifconfig eth0 up
 up ifup ppp0=dsl
 down ifdown ppp0=dsl
 post-down /sbin/ifconfig eth0 down
# Folgendes wird nur intern verwendet:
iface dsl inet ppp
 provider dsl-provider
```

5.5.12. Der Netzwerkkonfigurations-Status von ifupdown

Die Datei "**/etc/network/run/ifstate**" speichert die **vorgesehenen** Netzwerkkonfigurations-Status für alle derzeit aktiven, durch das **ifupdown**-Paket verwalteten Netzwerkschnittstellen. Unglücklicherweise listet die "**/etc/network/run/ifstate**"-Datei eine Schnittstelle selbst dann als aktiv auf, wenn diese von **ifupdown** nicht aktiviert werden konnte.

Solange die Ausgabe des Befehls ifconfig(8) für eine Schnittstelle keine Zeile wie in folgenden Beispiel enthält, kann die Schnittstelle nicht als Teil eines IPV4-Netzwerks genutzt werden:

```
  inet addr:192.168.11.2  Bcast:192.168.11.255  Mask:255.255.255.0
```

Anmerkung

Für das Ethernet-Gerät, das mittels PPPoE verbunden ist, enthält die ifconfig(8)-Ausgabe keine solche Zeile wie in obigem Beispiel.

5.5.13. Die grundlegende Netzwerk-Neukonfiguration

Wenn Sie versuchen, eine Schnittstelle wie z.B. **eth0** neu zu konfigurieren, müssen Sie sie zunächst mit dem Befehl "**sudo ifdown eth0**" deaktivieren. So wird der Eintrag für **eth0** aus der "**/etc/network/run/ifstate**"-Datei entfernt. (Dies könnte zu Fehlermeldungen führen, falls **eth0** nicht aktiviert ist oder vorher unpassend konfiguriert war. Allerdings ist es offensichtlich unproblematisch, dies auf einfachen Arbeitsplatzsystemen für Einzelbenutzer so durchzuführen.)

Sie können jetzt die Inhalte von "**/etc/network/interfaces**" wie für die Neukonfiguration der Schnittstelle **eth0** erforderlich anpassen.

Aktivieren Sie dann wieder die **eth0**-Schnittstelle mit dem Befehl "**sudo ifup eth0**".

Tipp

Sie können eine einfache Neuinitialisierung der Netzwerkschnittstelle mittels "**`sudo ifdown eth0;sudo ifup eth0`**" durchführen.

5.5.14. Das ifupdown-extra-Paket

Das Paket **`ifupdown-extra`** bietet einfache Netzwerk-Verbindungstests zur Verwendung mit dem **`ifupdown`**-Paket:

- Der Befehl network-test(1) kann von der Shell aus genutzt werden.
- Die automatischen Skripte werden bei jeder **`ifup`**-Befehlsausführung gestartet.

Der **`network-test`**-Befehl befreit Sie von der Verwendung umständlicher Basisbefehle zur Analyse von Netzwerkproblemen.

Die automatischen Skripte werden in "**`/etc/network/*/`**" installiert und bieten Folgendes:

- prüfen der Verbindung des Netzwerkkabels;
- Überprüfung auf mehrfache Verwendung einer IP-Adresse;
- Einrichtung der statischen Routen des Systems basierend auf den Definitionen in "**`/etc/network/routes`**";
- Prüfung, ob der Netzwerk-Gateway erreichbar ist;
- Aufzeichnung der Ergebnisse in der Datei "**`/var/log/syslog`**".

Diese syslog-Aufzeichnungen sind für die Administration von Netzwerkproblemen auf fernen Systemen sehr nützlich.

Tipp

Das automatische Verhalten des **`ifupdown-extra`**-Pakets ist über "**`/etc/default/network-test`**" konfigurierbar. Einige dieser automatischen Tests können den Systemstart ein wenig verlangsamen, da die Antworten auf ARP-Anfragen eine gewisse Zeit benötigen.

5.6. Die erweiterte Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)

Die Funktionalität des **`ifupdown`**-Pakets kann mit entsprechendem fortgeschrittenen Wissen über das oben in Abschnitt 5.5, „Die grundlegende Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)“ beschriebene Maß hinaus erweitert werden.

Die hier beschriebenen Funktionalitäten sind absolut optional. Ich für meinen Teil, der ich faul und ein Minimalist bin, habe mich nur selten darum bemüht, sie zu nutzen.

Achtung

Falls Sie mit obigen Informationen aus Abschnitt 5.5, „Die grundlegende Netzwerkkonfiguration mit ifupdown (nach altbewährtem System)“ keine Netzwerkverbindung einrichten konnten, werden Sie mit den hier folgenden Informationen die Situation nur noch schlechter machen.

5.6.1. Das ifplugd-Paket

Das **`ifplugd`**-Paket ist ein älteres Werkzeug für automatische Netzwerkkonfiguration, das nur Ethernet-Verbindungen verwalten kann. Es behebt Probleme mit abgezogenen/neu eingesteckten Ethernet-Kabeln bei mobilen PCs usw. Falls Sie NetworkManager oder Wicd (lesen Sie dazu Abschnitt 5.2, „Die moderne Netzwerkkonfiguration für Arbeitsplatzsysteme“) installiert haben, benötigen Sie dieses Paket nicht.

Dieses Paket läßt einen Daemon laufen und ersetzt **auto**- oder **allow-hotplug**-Funktionalitäten (Näheres in Tabelle 5.10, „Liste der Einträge in "**/etc/network/interfaces**"“) und aktiviert entsprechende Schnittstellen bei deren Anschluß an das Netzwerk.

Hier eine Anleitung, wie Sie das **ifplugd**-Paket für die interne Ethernet-Schnittstelle, z.B. **eth0**, verwenden:

1. Entfernen Sie folgende Einträge in "**/etc/network/interfaces**": "**auto eth0**" oder "**allow-hotplug eth0**".
2. Folgende Einträge müssen in "**/etc/network/interfaces**" erhalten bleiben: "**iface eth0 inet …**" und "**mapping …**".
3. Installieren Sie das Paket **ifplugd**
4. Führen Sie "**sudo dpkg-reconfigure ifplugd**" aus.
5. Wählen Sie **eth0** bei der Abfrage "Zu überwachende Schnittstellen, die immer (statisch) vorhanden sind".

Jetzt funktioniert die Netzwerk-Neukonfiguration wie gewünscht.

- Nach dem Einschalten oder bei Erkennung neuer Hardware wird die Schnittstelle nicht selbstständig aktiviert.
  - Schneller Boot-Prozess ohne das lange Warten auf die DHCP-Zeitüberschreitung.
  - Keine seltsame Aktivierung von Schnittstellen ohne korrekte IPv4-Adresse (lesen Sie auch Abschnitt 5.5.12, „Der Netzwerkkonfigurations-Status von ifupdown“).
- Nach Erkennung des eingesteckten Ethernet-Kabels wird die Schnittstelle aktiviert.
- Einige Zeit nach dem Abziehen des Ethernet-Kabels wird die Schnittstelle automatisch deaktiviert.
- Nach dem Einstecken eines anderen Ethernet-Kabels wird die Schnittstelle (für die neue Netzwerkumgebung passend) aktiviert.

Tipp

Über die Argumente für den ifplugd(8)-Befehl kann dessen Verhalten, wie z.B. die Verzögerung bis zur Neukonfiguration der Schnittstelle, angepasst werden.

5.6.2. Das ifmetric-Paket

Das **ifmetric**-Paket ermöglicht es uns, die Metrik von Routen sogar für DHCP zu manipulieren.

Mit folgendem Vorgehen setzen Sie die **eth0**-Schnittstelle als bevorzugt gegenüber der **wlan0**-Schnittstelle:

1. Installieren Sie das Paket **ifmetric**.
2. Fügen Sie eine Zeile mit "**metric 0**" direkt unterhalb von "**iface eth0 inet dhcp**" in "**/etc/network/interfaces**" ein.
3. Fügen Sie eine Zeile mit "**metric 1**" direkt unterhalb von "**iface wlan0 inet dhcp**" in "**/etc/network/interfaces**" ein.

Das "metric 0" steht für die Route mit der höchsten Priorität und ist der Standardwert. Der größere metric-Wert steht für Routen mit niedrigeren Prioritäten. Die IP-Adresse der aktivierten Schnittstelle mit dem niedrigsten metric-Wert wird die bevorzugte Adresse. Details finden Sie unter ifmetric(8).

5.6.3. Die virtuelle Schnittstelle

Mit einer einzigen physikalischen Ethernet-Schnittstelle können mehrere virtuelle Schnittstellen mit verschiedenen IP-Adressen konfiguriert werden. Der übliche Ansatz ist hierbei, eine

Schnittstelle mit mehreren IP-Subnetzen zu verbinden. Zum Beispiel ist das IP-Adress-basierte virtuelle Web-Hosting mit einer einzigen Netzwerkschnittstelle eine solche Anwendung.

Gehen wir zum Beispiel einmal von folgendem aus:

- Eine einzelne Ethernet-Schnittstelle in Ihrem Rechner ist mit einem Ethernet-Hub verbunden (nicht mit dem Breitband-Router).
- Der Ethernet-Hub ist sowohl mit dem Internet wie auch mit dem LAN verbunden.
- Das LAN-Netzwerk nutzt das Subnetz **`192.168.0.x/24`**.
- Ihr Rechner nutzt eine per DHCP verwaltete IP-Adresse mit der physikalischen Schnittstelle **`eth0`** für das Internet.
- Der Rechner nutzt **`192.168.0.1`** mit der virtuellen Schnittstelle **`eth0:0`** für das LAN.

Folgende Einträge in "**`/etc/network/interfaces`**" konfigurieren Ihr Netzwerk:

```
iface eth0 inet dhcp
 metric 0
iface eth0:0 inet static
 address 192.168.0.1
 netmask 255.255.255.0
 network 192.168.0.0
 metric 1
```

Achtung

Obwohl dieses Konfigurationsbeispiel mit Network Address Translation (NAT) über netfilter/iptables (lesen Sie dazu Abschnitt 5.9, „Die Netfilter-Infrastruktur“) günstige Router für das LAN mit nur einer einzigen Schnittstelle ermöglicht, hat solch ein Setup keine wirklichen Firewall-Fähigkeiten. Sie sollten zwei physikalische Schnittstellen mit NAT nutzen, um das lokale Netzwerk gegen das Internet abzuschirmen.

5.6.4. Die erweiterte Befehlssyntax

Das **`ifupdown`**-Paket bietet erweiterte Netzwerkkonfigurationen über **Netzwerkkonfigurations**-Namen und **Netzwerkschnittstellen**-Namen. Ich verwende eine Bezeichnungsweise, die sich leicht von der in ifup(8) und interfaces(5) unterscheidet.

Tabelle 5.12. Liste verschiedener Bezeichnungsweisen für Netzwerkschnittstellen

Bezeichnung in der Handbuchseite	**Meine Bezeichnung**	**Beispiele im folgenden Text**	**Beschreibung**
Name der **physikalischen Schnittstelle**	Name der **Netzwerkschnittstelle**	**`lo`**, **`eth0`**, **`<schnittstellen-name>`**	Name, der durch den Linux-Kernel (mittels **udev**-Mechanismus) vergeben wird
Name der **logischen Schnittstelle**	Name der **Netzwerkkonfiguration**	**`config1`**, **`config2`**, **`<konfigurations-name>`**	Namensbezeichnung gemäß **`iface`** in "**`/etc/network/interfaces`**"

Die Befehle zur grundlegenden Netzwerkkonfiguration in Abschnitt 5.5.1, „Die Befehlssyntax in vereinfachter Form“ erfordern es, dass der **Netzwerkkonfigurations**-Name aus dem **`iface`**-Eintrag mit dem Name der **Netzwerkschnittstelle** in "**`/etc/network/interfaces`**" übereinstimmt.

Die erweiterten Befehle zur Netzwerkkonfiguration erlauben die Trennung von **Netzwerkkonfigurations**-Name und **Netzwerkschnittstellen**-Name in "**`/etc/network/interfaces`**" wie folgt:

Tabelle 5.13. Liste erweiterter Befehle zur Netzwerkkonfiguration mit ifupdown

Befehl	Aktion
ifup eth0=config1	die Netzwerkschnittstelle **eth0** mit der Konfiguration **config1** aktivieren
ifdown eth0=config1	die Netzwerkschnittstelle **eth0** mit der Konfiguration **config1** deaktivieren
ifup eth0	die Netzwerkschnittstelle **eth0** mit der über den **mapping**-Eintrag gewählten Konfiguration aktivieren
ifdown eth0	die Netzwerkschnittstelle **eth0** mit der über den **mapping**-Eintrag gewählten Konfiguration deaktivieren

5.6.5. Der mapping-Eintrag

Um Verwirrungen zu vermeiden, haben wir die Erklärung des **mapping**-Eintrags in der "**/etc/network/interfaces**"-Datei im Abschnitt 5.5.2, „Der grundlegende Aufbau von "/etc/network/interfaces"“ übersprungen. Dieser Eintrag hat folgende Syntax:

```
mapping <muster-für-schnittstellenname>
 script <skript-name>
 map <skript-eingabewert1>
 map <script-eingabewert2>
 map ...
```

Dieses ermöglicht erweiterte Funktionalitäten für die "**/etc/network/interfaces**"-Datei, indem die Wahl der Konfiguration über das mittels **<skript-name>** festgelegte mapping-Skript automatisiert wird.

Wir wollen einmal dieser Programmausführung folgen:

```
$ sudo ifup eth0
```

Wenn "**<muster-für-schnittstellenname>**" auf "**eth0**" zutrifft, wird automatisch die Ausführung des folgenden Befehls ausgelöst, um **eth0** zu konfigurieren:

```
$ sudo ifup eth0=$(echo -e '<skript-eingabewert1> \n <skript-eingabewert2>
\n ...' | <skript-name> eth0)
```

Hierbei sind die Zeilen mit "**map**" optional und können wiederholt verwendet werden.

Anmerkung

Das Muster für den **mapping**-Eintrag funktioniert wie die Shell-Dateinamen-Globs (Näheres in Abschnitt 1.5.6, „Shell-Glob“).

5.6.6. Die manuell umschaltbare Netzwerkkonfiguration

Hier erklären wir, wie Sie manuell zwischen mehreren Netzwerkkonfigurationen wechseln können, ohne die "**/etc/network/interfaces**"-Datei umzuschreiben (wie es in Abschnitt 5.5.13, „Die grundlegende Netzwerk-Neukonfiguration“ beschrieben ist).

Für alle Netzwerkkonfigurationen, auf die Sie zugreifen müssen, erstellen Sie separate Einträge in "**/etc/network/interfaces**" wie folgt:

```
auto lo
iface lo inet loopback

iface config1 inet dhcp

iface config2 inet static
 address 192.168.11.100
 netmask 255.255.255.0
 gateway 192.168.11.1
 dns-domain example.com
 dns-nameservers 192.168.11.1

iface pppoe inet manual
 pre-up /sbin/ifconfig eth0 up
 up ifup ppp0=dsl
 down ifdown ppp0=dsl
 post-down /sbin/ifconfig eth0 down

# Folgendes wird nur intern verwendet:
iface dsl inet ppp
 provider dsl-provider

iface pots inet ppp
 provider provider
```

Bitte beachten Sie, dass der **Netzwerkkonfigurations**-Name, also das Kürzel hinter **`iface`**, nicht identisch ist mit dem **Netzwerkschnittstellen**-Namen. Auch gibt es keine **`auto`**- oder **`allow-hotplug`**-Einträge, über die die Netzwerkschnittstelle **`eth0`** beim Auftreten entsprechender Ereignisse automatisch gestartet würde.

Jetzt sind Sie bereit für den Wechsel zwischen den verschiedenen Konfigurationen.

Verbinden Sie den PC mit einem per DHCP verwalteten LAN-Netzwerk. Sie aktivieren die **Netzwerkschnittstelle** (die physikalische Schnittstelle) **`eth0`**, indem Sie ihr wie folgt den **Netzwerkkonfigurations**-Namen (den logischen Schnittstellennamen) **`config1`** zuweisen:

```
$ sudo ifup eth0=config1
Password:
...
```

Die Schnittstelle **`eth0`** wird aktiviert, per DHCP konfiguriert und mit dem LAN verbunden.

```
$ sudo ifdown eth0=config1
...
```

Die Schnittstelle **`eth0`** wird deaktiviert und vom LAN getrennt.

Nun verbinden Sie den PC mit einem LAN-Netzwerk mit statischen IP-Adressen. Sie aktivieren die **Netzwerkschnittstelle `eth0`**, indem Sie ihr wie folgt den **Netzwerkkonfigurations**-Namen **`config2`** zuweisen:

```
$ sudo ifup eth0=config2
...
```

Die Schnittstelle **`eth0`** wird aktiviert, mit einer statischen IP-Adresse konfiguriert und mit dem LAN verbunden. Über die zusätzlichen **`dns-*`**-Parameter werden die Inhalte von "**`/etc/resolv.conf`**" festgelegt. Die Verwaltung von "**`/etc/resolv.conf`**" funktioniert aber besser, wenn das Paket **`resolvconf`** installiert ist.

```
$ sudo ifdown eth0=config2
...
```

Die Schnittstelle **`eth0`** wird erneut deaktiviert und vom LAN getrennt.

Nun verbinden Sie den PC mit einem Anschluß des Breitband-Modems, das per PPPoE bedient wird. Sie aktivieren die **Netzwerkschnittstelle `eth0`**, indem Sie ihr wie folgt den **Netzwerkkonfigurations**-Namen **pppoe** zuweisen:

```
$ sudo ifup eth0=pppoe
...
```

Die Schnittstelle **`eth0`** wird aktiviert und über die direkte PPPoE-Verbindung zum ISP konfiguriert.

```
$ sudo ifdown eth0=pppoe
...
```

Die Schnittstelle **`eth0`** wird erneut deaktiviert und vom LAN getrennt.

Zu guter Letzt verbinden Sie den PC ohne LAN oder Breitband-Modem via Modem und Telefonanschluß (POTS) mit dem Internet. Sie aktivieren die **Netzwerkschnittstelle `ppp0`**, indem Sie ihr wie folgt den **Netzwerkkonfigurations**-Namen **`pots`** zuweisen:

```
$ sudo ifup ppp0=pots
...
```

Die Schnittstelle **`ppp0`** wird aktiviert und via PPP mit dem Internet verbunden.

```
$ sudo ifdown ppp0=pots
...
```

Die Schnittstelle **`ppp0`** wird deaktiviert und die Verbindung zum Internet getrennt.

Sie sollten die "**`/etc/network/run/ifstate`**"-Datei bezüglich des aktuellen **`ifupdown`**-Netzwerkkonfigurations-Status kontrollieren.

> **Warnung**
>
> Falls Sie mehrere Netzwerkschnittstellen in Ihrem PC haben, müssen Sie unter Umständen die Zahlen von **`eth*`**, **`ppp*`** usw. anpassen.

5.6.7. Skriptverarbeitung mit dem ifupdown-System

Das **`ifupdown`**-System führt automatisch Skripte aus, die in "**`/etc/network/*/`**" installiert sind; dabei werden Umgebungsvariablen an die Skripte exportiert.

Tabelle 5.14. Liste der Umgebungsvariablen, die dem ifupdown-System übergeben werden

Umgebungsvariable	Übergebener Wert
"**`$IFACE`**"	physikalischer Name (Schnittstellenname) der zu verarbeitenden Schnittstelle
"**`$LOGICAL`**"	logischer Name (Konfigurationsname) der zu verarbeitenden Schnittstelle
"**`$ADDRFAM`**"	<addressfamilie> der Schnittstelle
"**`$METHOD`**"	<methodenname> der Schnittstelle (z.B. "statisch")
"**`$MODE`**"	"start", wenn von **`ifup`** ausgeführt; "stop", wenn von **`ifdown`** ausgeführt
"**`$PHASE`**"	wie "**`$MODE`**", aber mit feinerer Auflösung, unterscheidet die Phasen **`pre-up`**, **`post-up`**, **`pre-down`** und **`post-down`**
"**`$VERBOSITY`**"	gibt an, ob "**`--verbose`**" genutzt wird (ausführliche Protokollausgabe); Wert = 1 falls genutzt, Wert = 0 falls nicht
"**`$PATH`**"	Befehlssuchpfad: z.B. "**`/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin`**"
"**`$IF_<OPTION>`**"	Wert für die entsprechende Option im **`iface`**-Eintrag

Hierbei wird die Umgebungsvariable "**`$IF_<OPTION>`**" aus dem Namen für die zugehörige Option

(z.B. <option1> oder <option2>) gebildet; dazu wird "**$IF_**" vorangestellt, alle Buchstaben werden in Großbuchstaben umgewandelt, Bindestriche in Unterstriche geändert und nicht-alphanumerische Zeichen werden entfernt.

Tipp

Lesen Sie Abschnitt 5.5.2, „Der grundlegende Aufbau von "/etc/network/interfaces"“ bezüglich weiterer Details zu <addressfamilie>, <methodenname>, <option> usw.

Das **ifupdown-extra**-Paket (Näheres in Abschnitt 5.5.14, „Das ifupdown-extra-Paket“) verwendet diese Umgebungsvariablen, um die Funktionalität des **ifupdown**-Pakets zu erweitern. Das Paket **ifmetric** (lesen Sie dazu Abschnitt 5.6.2, „Das ifmetric-Paket“) installiert das Skript "**/etc/network/if-up.d/ifmetric**", welches über die Variable "**$IF_METRIC**" die Metric festlegt. Auch das Paket **guessnet** (weitere Infos dazu in Abschnitt 5.6.8, „Mapping mit guessnet“), das ein einfaches und leistungsfähiges Rahmenwerk zur automatischen Auswahl der Netzwerkkonfiguration über den mapping-Mechanismus bietet, nutzt diese Variablen.

Anmerkung

Weiterführende Beispiele für speziell angepasste Netzwerkkonfigurations-Skripte mittels dieser Umgebungsvariablen finden Sie in "**/usr/share/doc/ifupdown/examples/***" sowie in den Paketen **ifscheme** und **ifupdown-scripts-zg2**. Diese zusätzlichen Skripte haben einige Überschneidungen mit den Funktionalitäten aus den Paketen **ifupdown-extra** und **guessnet**. Falls Sie diese zusätzlichen Skripte installieren, sollten Sie vorgenannte Skripte anpassen, um wechselseitige Störungen zu vermeiden.

5.6.8. Mapping mit guessnet

Statt der in Abschnitt 5.6.6, „Die manuell umschaltbare Netzwerkkonfiguration“ beschriebenen manuellen Auswahl der Konfiguration können Sie auch den Mapping-Mechanismus (Näheres in Abschnitt 5.6.5, „Der mapping-Eintrag“) verwenden, um die Netzwerkkonfiguration mit speziell auf Ihre Situation angepassten Skripten automatisch auszuwählen.

Der Befehl guessnet-ifupdown(8) aus dem **guessnet**-Paket wurde als Mapping-Skript entwickelt und bietet ein leistungsfähiges Rahmenwerk zum **ifupdown**-System.

- Sie legen für jeden **iface**-Eintrag Testbedingungen als Wert für die **guessnet**-Option fest.
- Der Mapping-Mechanismus wählt den ersten **iface**-Eintrag als Netzwerkkonfiguration aus, für den die Abfrage kein ERROR-Ergebnis (Fehler) zurückgibt.

Diese doppelte Verwendung der "**/etc/network/interfaces**"-Datei, einmal durch das mapping-Skript (**guessnet-ifupdown**) und dann noch durch die originale Netzwerk-Infrastruktur (**ifupdown**) hat keine negativen Nebenwirkungen, da durch die **guessnet**-Optionen nur zusätzliche Umgebungsvariablen an die vom **ifupdown**-System ausgeführten Skripte exportiert werden. Details dazu finden Sie in guessnet-ifupdown(8).

Anmerkung

Falls mehrere **guessnet**-Optionen in "**/etc/network/interfaces**" benötigt werden, verwenden Sie Optionen, die mit **guessnet1**, **guessnet2** usw. beginnen, da das **ifupdown**-Paket es nicht erlaubt, die gleiche Zeichenkette am Anfang von Optionszeilen mehrfach zu verwenden.

5.7. Die Netzwerkkonfiguration auf unterster Ebene

5.7.1. iproute2-Befehle

Die iproute2-Befehle bieten vollwertige Funktionalität auf der untersten Ebene der Netzwerkkonfiguration. Hier eine Tabelle zur Gegenüberstellung von veralteten net-tools-Befehlen

und neuen iproute2- und anderen Befehlen.

Tabelle 5.15. Gegenüberstellung von `net-tools`- und `iproute2`-Befehlen

net-tools (veraltet)	iproute2 usw. (neu)	Beeinflussung
ifconfig(8)	**`ip addr`**	Protokoll-Adresse (IP oder IPv6) eines Gerätes
route(8)	**`ip route`**	Eintrag in der Routing-Tabelle
arp(8)	**`ip neigh`**	ARP- oder NDISC-Cache-Eintrag
`ipmaddr`	**`ip maddr`**	Multicast-Adresse
`iptunnel`	**`ip tunnel`**	Tunnel über IP
nameif(8)	ifrename(8)	Netzwerkschnittstellen basierend auf MAC-Adressen benennen
mii-tool(8)	ethtool(8)	Einstellungen von Ethernet-Geräten

Lesen Sie ip(8) und das IPROUTE2 Utility Suite Howto.

5.7.2. Sichere Basis-Netzwerkoperationen

Sie können die folgenden Netzwerkbefehle der untersten Ebene problemlos verwenden, da sie die Netzwerkkonfiguration nicht verändern:

Tabelle 5.16. Liste von Basis-Netzwerkbefehlen

Befehl	Beschreibung
`ifconfig`	Verbindungs- und Adressstatus von aktiven Schnittstellen anzeigen
`ip addr show`	Verbindungs- und Adressstatus von aktiven Schnittstellen anzeigen
`route -n`	Vollständige Routing-Tabelle mit numerischen Adressen anzeigen
`ip route show`	Vollständige Routing-Tabelle mit numerischen Adressen anzeigen
`arp`	Aktuellen Inhalt der ARP-Cache-Tabellen anzeigen
`ip neigh`	Aktuellen Inhalt der ARP-Cache-Tabellen anzeigen
`plog`	Logdaten des PPP-Daemons anzeigen
`ping yahoo.com`	Internet-Verbindung zu "**`yahoo.com`**" überprüfen
`whois yahoo.com`	Überprüfen, wer "**`yahoo.com`**" in der Domain-Datenbank registriert hat
`traceroute yahoo.com`	Verbindung zu "**`yahoo.com`**" durch das Internet verfolgen
`tracepath yahoo.com`	Verbindung zu "**`yahoo.com`**" durch das Internet verfolgen
`mtr yahoo.com`	Verbindung zu "**`yahoo.com`**" durch das Internet verfolgen (wiederholt)
`dig [@dns-server.com] example.com [{a|mx|any}]`	DNS-Einträge von "**`example.com`**" laut den Daten von "**`dns-server.com`**" auf einen "**`a`**"-, "**`mx`**"- oder "**`any`**"-Eintrag überprüfen
`iptables -L -n`	Paketfilter überprüfen
`netstat -a`	Alle offenen Ports finden
`netstat -l --inet`	Ports finden, die auf eine Verbindung warten
`netstat -ln --tcp`	TCP-Ports finden, die auf eine Verbindung warten (numerisch)
`dlint example.com`	DNS-Zonen-Informationen von "**`example.com`**" überprüfen

Tipp

Einige dieser Basisbefehle zur Netzwerkkonfiguration sind in "**`/sbin/`**" abgelegt. Sie müssen unter Umständen den vollständigen Pfad, wie z.B.

"**/sbin/ifconfig**" angeben oder "**/sbin**" zur Variable "**$PATH**" in Ihrer "**~/.bashrc**"-Datei hinzufügen.

5.8. Netzwerkoptimierung

Die grundsätzliche Netzwerkoptimierung liegt außerhalb des Rahmens dieser Dokumentation. Ich erwähne hier nur Dinge, die für Anwender-typische Verbindungen passend sind.

Tabelle 5.17. Liste von Werkzeugen zur Netzwerkoptimierung

Pakete	Popcon	Größe	Beschreibung
iftop	V:10, I:105	93	Informationen zur Bandbreitennutzung einer Netzwerkschnittstelle anzeigen
iperf	V:5, I:53	149	Werkzeug zur IP-Bandbreiten-Messung
apt-spy	V:0, I:6	105	Eine "**/etc/apt/sources.list**"-Datei basierend auf Bandbreitentests erstellen
ifstat	V:1, I:10	54	InterFace STATistics Monitoring (Netzwerkschnittstellen-Statistik/-Überwachung)
bmon	V:1, I:11	138	Portierbarer Bandbreitenmonitor und Geschwindigkeitsrechner
ethstatus	V:0, I:4	38	Skript, das schnell den Durchsatz eines Netzwerkgerätes messen kann
bing	V:0, I:2	57	Empirisch stochastischer Bandbreitentester
bwm-ng	V:1, I:16	89	Kleiner und einfacher konsolenbasierter Bandbreitenmonitor
ethstats	V:0, I:1	20	Konsolenbasierter Ethernet-Statistikmonitor
ipfm	V:0, I:0	156	Bandbreitenanalyse-Werkzeug

5.8.1. Die optimale MTU finden

Der Wert für die Maximum Transmission Unit (MTU) kann experimentell mit ping(8) mittels der Option "**-M do**" ermittelt werden; dabei werden ICMP-Pakete mit einer Datengröße ab 1500 (mit einem Offset von 28 Byte für den IP+ICMP-Header) versendet, um herauszufinden, welche maximale Größe ohne IP-Fragmentierung möglich ist.

Probieren Sie zum Beispiel folgendes:

```
$ ping -c 1 -s $((1500-28)) -M do www.debian.org
PING www.debian.org (194.109.137.218) 1472(1500) bytes of data.
From 192.168.11.2 icmp_seq=1 Frag needed and DF set (mtu = 1454)

--- www.debian.org ping statistics ---
0 packets transmitted, 0 received, +1 errors
```

Versuchen Sie dann 1454 statt 1500.

Sie sehen, dass ping(8) bei einem Wert von 1454 erfolgreich beendet wird.

Dies ist das Path MTU (PMTU) Discovery-Verfahren (RFC1191) und der Befehl tracepath(8) kann dies automatisieren.

Tipp

Das obige Beispiel mit einem PMTU-Wert von 1454 stammt von meinem früheren FTTP-Provider, der Asynchronous Transfer Mode (ATM) als Backbone-Netzwerk verwendet und seine Kunden per PPPoE bedient. Der bei Ihnen aktuelle PMTU-Wert hängt von der jeweiligen Situation ab, z.B. habe ich bei meinem neuen FTTP-Provider einen Wert von 1500.

Tabelle 5.18. Wesentliche Grundregeln für den optimalen MTU-Wert

Netzwerkumgebung	MTU	Argumentation

Netzwerkumgebung	MTU	Argumentation
Einwahlverbindung (IP: PPP)	576	Standard
Ethernet-Verbindung (IP: DHCP oder fest)	1500	Standard und vorgegeben
Ethernet-Verbindung (IP: PPPoE)	1492 (=1500-8)	2 Byte für den PPP-Header und 6 Byte für den PPPoE-Header
Ethernet-Verbindung (Backbone des ISP: ATM. IP: DHCP oder fest)	1462 (=48*31-18-8)	Spekulation des Autors: 18 Byte für den Ethernet-Header, 8 Byte für den SAR-Trailer
Ethernet-Verbindung (Backbone des ISP: ATM. IP: PPPoE)	1454 (=48*31-8-18-8)	Unter "Optimal MTU configuration for PPPoE ADSL Connections" finden Sie nähere Infos.

Zusätzlich zu diesen Grundregeln sollten Sie folgendes wissen:

- Jegliche Nutzung von Tunneling-Methoden (VPN usw.) kann aufgrund des Overheads den optimalen MTU-Wert reduzieren.
- Der MTU-Wert sollte den über die experimentelle Methode ermittelten PMTU-Wert nicht überschreiten.
- Ein größerer MTU-Wert ist grundsätzlich besser, wenn andere Einschränkungen greifen.

5.8.2. Setzen der MTU

Hier einige Beispiele, wie Sie den MTU-Wert von seinem Standard 1500 auf 1454 setzen:

Für DHCP (Näheres in Abschnitt 5.5.4, „Die über DHCP bediente Netzwerkschnittstelle“) können Sie den entsprechenden **iface**-Eintrag in "**/etc/network/interfaces**" wie folgt ersetzen:

```
iface eth0 inet dhcp
 pre-up /sbin/ifconfig $IFACE mtu 1454
```

Bei Verwendung statischer IPs (wie in Abschnitt 5.5.5, „Die Netzwerkschnittstelle mit statischer IP-Adresse“) können Sie den entsprechenden **iface**-Eintrag in "**/etc/network/interfaces**" wie folgt anpassen:

```
iface eth0 inet static
 address 192.168.11.100
 netmask 255.255.255.0
 gateway 192.168.11.1
 mtu 1454
 dns-domain example.com
 dns-nameservers 192.168.11.1
```

Bei Verwendung von direktem PPPoE (wie in Abschnitt 5.4.5, „Die PPPoE-Verbindung mit pppoeconf“) können Sie die entsprechende "**mtu**"-Zeile in "**/etc/ppp/peers/dsl-provider**" durch folgendes ersetzen:

```
mtu 1454
```

Die Maximum Segment Size (MSS) wird als alternative Messmethode für die Paketgröße verwendet. Der Zusammenhang zwischen MSS und MTU ist wie folgt:

- MSS = "MTU - 40" bei IPv4
- MSS = "MTU - 60" bei IPv6

Anmerkung

Bei Netzwerkoptimierung mittels iptables(8) (lesen Sie dazu auch Abschnitt 5.9, „Die Netfilter-Infrastruktur“) kann die Paketgröße über die MSS begrenzt werden; dies ist für einen Router nützlich. Lesen Sie den Abschnitt bezüglich "TCPMSS" in iptables(8).

5.8.3. WAN-TCP-Optimierung

Der TCP-Durchsatz kann über die Anpassung von Parametern zur TCP-Puffergröße maximiert werden, wie es die Anleitungen "TCP Tuning Guide" und "TCP Tuning" für modernes WAN mit hoher Bandbreite und hoher Latenz beschreiben. Das soll hierzu genügen; die aktuellen Debian-Standardeinstellungen funktionieren für mein LAN mit seiner Verbindung zum sehr schnellen 1G bps FFTP-Dienst sehr gut.

5.9. Die Netfilter-Infrastruktur

Netfilter stellt eine Infrastruktur für Stateful Packet Inspection (SPI, zustandsorientierte Paketüberprüfung) und Network Address Translation (NAT) über Module des Linux-Kernels (lesen Sie hierzu Abschnitt 3.3.1, „Die Kernel-Modul-Initialisierung“) zur Verfügung.

Tabelle 5.19. Liste von Firewall-Werkzeugen

Pakete	Popcon	Größe	Beschreibung
iptables	V:246, I:996	1558	Administrationswerkzeuge für netfilter (iptables(8) für IPv4, ip6tables(8) für IPv6)
arptables	V:0, I:2	124	Administrationswerkzeuge für netfilter (arptables(8) für ARP)
ebtables	V:31, I:56	354	Administrationswerkzeuge für netfilter (ebtables(8) für Ethernet-Bridging-Betrieb)
iptstate	V:0, I:4	111	Fortlaufende Überwachung des netfilter-Status (ähnlich zu top(1))
shorewall-init	V:0, I:0	55	Initialisierung der Shoreline Firewall
shorewall	V:7, I:17	2191	Erzeugung von netfilter-Konfigurationsdateien für Shoreline Firewall
shorewall-lite	V:0, I:0	76	Erzeugung von netfilter-Konfigurationsdateien für Shoreline Firewall (abgespeckte Version)
shorewall6	V:1, I:2	858	Erzeugung von netfilter-Konfigurationsdateien für Shoreline Firewall (IPv6-Version)
shorewall6-lite	V:0, I:0	75	Erzeugung von netfilter-Konfigurationsdateien für Shoreline Firewall (abgespeckte IPv6-Version)

Das vorherrschende Nutzerprogramm für netfilter ist iptables(8). Sie können netfilter von Hand interaktiv über die Shell konfigurieren, seinen Status mit iptables-save(8) sichern und beim Systemstart über ein Init-Skript mittels iptables-restore(8) wiederherstellen.

Konfigurations-Hilfsskripte wie shorewall vereinfachen diesen Prozess.

Sie finden Dokumentation unter http://www.netfilter.org/documentation/ oder in "**/usr/share/doc/iptables/html/**":

- Linux Networking-Concepts HOWTO
- Linux 2.4 Packet Filtering HOWTO
- Linux 2.4 NAT HOWTO

Tipp

Obwohl für Linux **2.4** geschrieben, sind sowohl der iptables(8)-Befehl wie auch die Netfilter-Kernel-Funktionalität für die Linux-Kernel-Serien **2.6** und **3.x** passend.

Kapitel 4. Authentifizierung


Kapitel 6. Netzwerkapplikationen