

5.1 Netzwerk/Grundkonfiguration

5.1.1 Treiber und Netzwerkkarte, udev

Zur Benutzung der Netzwerkkarte wird ein Treiber benötigt, der mit der Karte auf die richtige Weise kommuniziert.

Ob dieser Treiber vorhanden ist und richtig arbeitet, erkennt man zuerst mit dem Befehl `ifconfig`¹²:

```
Terminal
schueler@debian964:~$ /sbin/ifconfig
eth0  Link encap:Ethernet  Hardware-Adresse 11:22:33:44:55:66
...
```

Es werden alle gefundenen Netzwerkgeräte angezeigt. Weitere Information erhält man noch mit:

```
Terminal
schueler@debian964:~$ ethtool -i eth0
driver: 8139too
...
```

Für WLAN-Karten gibt es noch spezielle Programme, die Informationen über die Funkverbindung herausgeben:

```
Terminal
schueler@debian964:~$ iwconfig wlan0
wlan0 IEEE 802.11bg  ESSID:"generic"
      Mode:Managed Frequency:2.437 GHz Access Point: 22:33:44:55:66:77
...
```

Weitere Programme dieser Art sind `iwlist` und `iwgetid` aus dem Debian-Paket `wireless-tools`.

Die Namen der Geräte haben eine Bedeutung:

- `lo` – Loopback-Gerät, virtuelles Netzwerkgerät
- `eth0` – erste Ethernet-Karte
- `wlan0` – erste WLAN-Karte

Wenn die Karte nicht erkannt wurde, muss man sich darum kümmern, ob ein passender Treiber existiert und wie man ihn einbindet.

Bei einem monolithischen Kernel ist der benötigte Treiber bereits eincompiliert worden und ist entweder aktiv oder nicht. Bei einem modularen Kernel (bei Linux der Normalfall) wird der Treiber, sobald eine Hardware erkannt wird, als so genanntes Modul eingebunden.

Man kann so ein Modul mit dem Befehl `modprobe` auch von Hand einbinden. Mit `lsmod` kann man sich die geladenen Module ansehen:

```
Terminal
root@debian964:~# modprobe 8139too
root@debian964:~# lsmod | grep 8139too
8139too          15109  0
mii              2714   2 8139too,8139cp
```

Man sieht hier, dass es einen Treiber `mii` gibt, auf dem zwei andere Treiber `8139too` und `8139cp` aufbauen. Mit `rmmod` kann man einen geladenen Treiber wieder entfernen. In der Datei `/etc/modules` (oder `conf.modules` oder `modules.conf`) ist festgehalten, welche Module beim Systemstart zusätzlich zu den vom Kernel bestimmten eingebunden werden müssen. Außerdem kann man in dieser Datei Optionen angeben.

¹Es ist vergleichbar mit dem Programm `ipconfig.exe` bei MS-Wind.

²Daneben gibt es noch den neueren Befehl `ip link`

5.1.2 Gerätenamen

Wenn ein Netzwerkgerät erkannt und der Treiber geladen wurde, dann muss das erkannte Gerät einen Namen bekommen, mit dem es von Dienstprogrammen aus benutzt werden kann.

Anwenderprogramme (Clients und Server) benutzen diesen Namen in der Regel *nicht*, sondern sie wählen ihren Kommunikationspartner über die IP-Adresse und ggf. die Portnummer (wobei der Benutzer stattdessen oft DNS-Name und Servicename eingeben kann). Deshalb sind Netzwerkgeräte in den allermeisten Fällen nicht über eine Gerätedatei im Dateisystem abgebildet³.

Wie findet man nun die Namen der Netzwerkgeräte? Als Nutzer sieht man sie in `/proc/net/dev`:

```
Terminal
schueler@debian964:~$ cat /proc/net/dev | tail -n +3 | cut -d: -f1
eth0
lo
```

Alternativ kann man auch das Programm `ifconfig` verwenden:

```
Terminal
schueler@debian964:~$ /sbin/ifconfig
eth0: flags=....
...
lo: flags=....
...
```

Das traditionelle Schema für die Benennung der Netzwerkgeräte ist wie folgt:

```
geraeteart nummer
```

Die Geräteart ist `eth` für Ethernet und `wlan` für WLAN. Weitere Arten siehe Tabelle 1. Die

Geräteart	Benennung	Beispiel
Loopback	lo (ohne Nummer)	lo
Ethernet	eth	eth0
WLAN	wlan	wlan0
Bluetooth	bt	bt0

Tabelle 1: Arten von Netzwerkgeräten und ihre Benennung

Nummern werden beginnend mit null fortlaufend vergeben.

Man kann auf einem Netzwerkgerät eine weitere Schnittstelle einrichten mit einer anderen Konfiguration. Das Namensschema für die zweite Schnittstelle auf demselben Gerät sieht dann so aus:

```
geraeteart nummer doppelstrich unternummer
```

Eine zweite Schnittstelle auf dem Gerät `eth0` hätte dann damit den Namen `eth0:1`.

Im Laufe der Zeit hat es sich als unpraktisch herausgestellt, die Nummern fortlaufend zu vergeben. Durch Hinzufügen von USB-Netzwerkgeräten konnte die Reihenfolge der Gerätenamen verschoben werden.

Um dieses Problem zu lösen, hat man das `udev`-System auch für Netzwerkkarten benutzt. Das `udev`-System dient eigentlich dazu, automatisch Gerätedateien für Massenspeicher anzulegen. Sobald ein USB-Memory-Stick eingesteckt wird, wird die `udev`-Datenbank nach passenden Einträgen durchsucht und daraufhin die entsprechende Gerätedatei im Verzeichnis `/dev` angelegt. Bei Netzwerkgeräten funktioniert das ebenso mit dem Unterschied, dass eben keine Gerätedatei angelegt wird, sondern ein Geräte name vergeben wird.

³Auf zeichenorientierte Geräte kann man direkt schreiben oder von ihnen lesen; das ist bei einer Netzwerkkarte unpraktisch; denn wohin soll sie schreiben? Blockorientierte Geräte kann man als Dateisystem einbinden. Auch dies ist für Netzwerkgeräte schlecht vorstellbar.

Später (mit systemd V.197, bei Debian ab Version 9) hat man sich dazu entschlossen, Geräte einfach nach ihrem Hardware-Anschluss zu benennen. Das neue Benennungsschema sieht aus wie folgt:

```
art namensart nummer
art namensart nummern-buchstaben-kombination
```

Die Geräteart findet man in Tabelle 2. Die Namensart ist aus Tabelle 3 zu entnehmen. Weitere

Geräteart	Benennung	Beispiel
Loopback	lo	lo
Ethernet	en	eno0
WLAN	wl	wlo0
WWAN	ww	ww00
Seriell	sl	slo0

Tabelle 2: Neue Benennung von Netzwerkgeräten, Geräteart

Namensart	Benennung	Beispiel
On-Board-Gerät	o	eno1
PCI-Express-Slot	s	ens1
Physikalischer Ort	p	enp2s0
MAC-Adresse	x	enx112233445566

Tabelle 3: Neue Benennung von Netzwerkgeräten, Namensart

Information findet man unter:

www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/.

5.1.3 Festlegen einer IP-Adresse

Zum einmaligen Konfigurieren einer Netzwerkkarte mit einer IP-Adresse samt Zubehör wie Netzwerkmaske usw. diente seit jeher das Programm `ifconfig` (seit einiger Zeit kann man dazu auch das neuere Programm `ip` verwenden):

```
Terminal
root@debian964:~# ifconfig eth0 192.168.2.1 netmask 255.255.255.0 up
```

Ist die Konfiguration in Ordnung, kann man sie in der Datei `/etc/network/interfaces` eintragen⁴. Dessen Inhalt wird beim Start vom Programm `ifup` ausgewertet und die entsprechende Aktion ausgeführt:

```
Terminal
/etc/network/interfaces
1 allow-hotplug eth0
2 iface eth0 inet static
3     address 192.168.2.1
4     netmask 255.255.255.0
```

Damit ist die Konfiguration dauerhaft gespeichert. Nun kann man `eth0` wie in der Datei beschrieben einschalten:

```
Terminal
root@debian964:~# ifup eth0
```

Ebenso kann man `eth0` ausschalten:

⁴Das Dateiformat kann man sich mit `man interfaces` ansehen.

```
Terminal
root@debian964:~# ifdown eth0
```

Beim Systemstart (oder beim Start des Netzwerks) schaltet man alle mit `auto` oder `allow-hotplug` gekennzeichneten Geräte ein:

```
Terminal
root@debian964:~# ifup -a
```

Beim Halt oder Reset (oder beim Stop des Netzwerks) schaltet man dieselben Geräte wieder aus:

```
Terminal
root@debian964:~# ifdown -a
```

Die Ähnlichkeit zur Konfiguration mit `/etc/fstab` und zu den Befehlen `mount` und `umount` ist deutlich.

Will man die Netzwerkkarte einmalig mit DHCP konfigurieren, kann man den Befehl `dhclient` benutzen:

```
Terminal
root@debian964:~# dhclient eth0
```

Falls sich das als erfolgreich herausstellt, kann man dies wieder in der Datei `/etc/network/interfaces` eintragen.

`/etc/network/interfaces`

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

5.1.4 Festlegen von Routen

Mit dem Programm `route` kann man bestimmen, welche Systeme über die Netzwerkkarte erreichbar sein sollen (neueres Programm: `ip route`). Zuerst wird die aktuelle Routing-Tabelle des Kernels angezeigt:

```
Terminal
root@debian964:~# route -n
Ziel          Router        Genmask       Flags Metric Ref  Use  Iface
192.168.2.0   0.0.0.0       255.255.255.0 U        0      0    0   eth0
192.168.1.0   0.0.0.0       255.255.255.0 U        0      0    0   wlan0
0.0.0.0       192.168.1.1   0.0.0.0       UG       0      0    0   wlan0
```

Nun wird eine Route gelöscht:

```
Terminal
root@debian964:~# route del -net 192.168.2.0/24
root@debian964:~# route -n
Ziel          Router        Genmask       Flags Metric Ref  Use  Iface
192.168.1.0   0.0.0.0       255.255.255.0 U        0      0    0   wlan0
0.0.0.0       192.168.1.1   0.0.0.0       UG       0      0    0   wlan0
```

Jetzt wird die gelöschte Route etwas anders eingerichtet und eine weitere Route über ein Gateway gelegt:

```
Terminal
root@debian964:~# route add -net 192.168.2.0/26 eth0
root@debian964:~# route add -net 192.168.3.0/24 gw 192.168.2.2
root@debian964:~# route -n
Ziel          Router        Genmask       Flags Metric Ref  Use  Iface
```

192.168.3.0	192.168.2.2	255.255.255.0	UG	0	0	0	eth0
192.168.2.0	0.0.0.0	255.255.255.192	U	0	0	0	eth0
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	wlan0
0.0.0.0	192.168.1.1	0.0.0.0	UG	0	0	0	wlan0

Weitere Möglichkeiten entnimmt man der Manual-Seite zu `route`. Nun kann man evtl. mit `ping` versuchen, andere Systeme zu erreichen⁵. Ein weiteres Programm zu diesem Zweck ist `traceroute`⁶.

5.1.5 ARP (und RARP)

Bekanntermaßen sorgen die Protokolle ARP und RARP für die Umsetzung zwischen 32-Bit-IP-Adresse und 48-Bit-MAC-Adresse, wobei ARP zu einer IP-Adresse die entsprechende MAC-Adresse finden soll und RARP in Gegenrichtung tätig ist. Beim ersten Aufruf einer IP-Adresse im lokalen Netz findet eine ARP-Anfrage statt, dessen Ergebnis dann im ARP-Cache gespeichert wird.

Den aktuellen ARP-Cache kann man sich anzeigen lassen mit folgendem Befehl:

```

Terminal
root@debian964:~# arp -n
Address          HWtype HWaddress          Flags Mask Iface
10.0.2.3 ether    52:55:0a:00:02:03 C          eth0
10.0.2.2 ether    52:55:0a:00:02:02 C          eth0

```

Mit der `-d`-Option kann man Einträge löschen und mit `-s` neue setzen:

```

Terminal
root@debian964:~# arp -d 10.0.2.2
root@debian964:~# arp -n
Address HWtype HWaddress          Flags Mask Iface
10.0.2.3 ether    52:55:0a:00:02:03 C          eth0
10.0.2.2      (unvollständig)          eth0
root@debian964:~# arp -s 10.0.2.2 52:55:0a:00:46:30
root@debian964:~# arp -n
Address HWtype HWaddress          Flags Mask Iface
10.0.2.3 ether    52:55:0a:00:02:03 C          eth0
10.0.2.3 ether    52:55:0a:00:46:30 CM         eth0

```

Die Größe des ARP-Caches kann man mit drei System-Variablen beeinflussen: `gc_thresh1`, `gc_thresh2` und `gc_thresh3`: Mindest-, Höchst- und absolute Maximalzahl von Einträgen⁷. Man kann sie sich anzeigen lassen mit dem Programm `sysctl`:

```

Terminal
root@debian964:~# sysctl | grep gc_thresh1
net.ipv4.neigh.default.gc_thresh1 = 128

```

Alternativ sind sie auch zu sehen unter dem `/proc/-Dateisystem`:

```

Terminal
root@debian964:~# cat /proc/sys/net/ipv4/neigh/default/gc_thresh1
128

```

Die Änderung geschieht wieder mit dem Programm `sysctl`, diesmal mit der `-w`-Option:

⁵Bei QEMU mit `user-net`-Konfiguration ist das nicht möglich, obwohl die Außenwelt – außer dem Host-System – komplett erreichbar ist

⁶Ja, es gibt Systeme, auf denen die gleiche Funktion `tracert.exe` heißt

⁷von englisch *threshold* = Schwelle

```

Terminal
root@debian964:~# sysctl -w net.ipv4.neigh.default.gc_thresh1=512
root@debian964:~# sysctl -w net.ipv4.neigh.default.gc_thresh1=1024
root@debian964:~# sysctl -w net.ipv4.neigh.default.gc_thresh1=2048

```

Um diese Änderungen dauerhaft zu speichern, muss man sie in der Datei `/etc/sysctl.conf` eintragen.

Das RARP-Protokoll soll es erlauben, dass ein System anhand seiner MAC-Adresse nach seiner IP fragt. Diese Funktionalität wird heutzutage durch seine Nachfolger BOOTP und DHCP abgedeckt.

5.1.6 Hostname und Domainname

Für die Ermittlung des eigenen Hostnamens und des eigenen Domainnamens gibt es zwei Befehle:

```

Terminal
root@debian964:~# hostname
debian964
root@debian964:~# dnsdomainname
schule.local

```

Der Hostname wird aus der Datei `/etc/hostname` entnommen⁸. Soll er geändert werden, muss man einfach diese Datei editieren. Der DNS-Domainname wird oft aus dem Domainanteil des vollen Hostnamens (FQDN) in `/etc/hosts` entnommen (je nach DNS-Konfiguration kann es auch anders sein).

5.1.7 DNS-Namensauflösung

Die Quellen der DNS-Auflösung und ihre Reihenfolge wird in der Datei `/etc/nsswitch.conf` festgelegt. Der wichtigste Eintrag lautet:

```

/etc/nsswitch.conf
1 hosts:          files dns

```

Das bedeutet: Bei der Suche nach Hostnamen soll zuerst in einer Datei (mit dem Namen `/etc/hosts`) gesucht werden; anschließend sollen die DNS-Server befragt werden (ihre IP-Adressen stehen in `/etc/resolv.conf`). Die erstgenannte Datei nennt eine IP-Adresse pro Zeile:

```

/etc/hosts
1 127.0.0.1        localhost debian964
2 10.1.45.198     debian964.schule.local  debian964
3 10.1.1.1        s1.schule.local  s1
4 10.1.1.2        s2.schule.local  s2
5 127.0.0.1        nervende.werbung.com

```

Diese Datei kann man durch eigene Einträge ergänzen (siehe letzte Zeile).

Die andere Datei nennt einen DNS-Server pro Zeile:

```

/etc/resolv.conf
1 search          schule.local
2 nameserver     10.1.1.3
3 nameserver     8.8.8.8

```

⁸Es gibt noch eine Datei `/etc/mailname` – sie enthält *den* Hostnamen, der bei der Mail-Zustellung verwendet wird.

Zur Erhöhung der Geschwindigkeit der DNS-Abfragen wird häufig ein DNS-Cache verwendet. Das Programm `nscd` (*name service caching demon*) stellt ihn zur Verfügung.

Mit dem Programm `host`⁹ kann man die Namensauflösung per DNS testen; der Inhalt der Datei `/etc/hosts` wird dabei nicht berücksichtigt:

```
Terminal
schueler@debian964:~$ host localhost
localhost has address 127.0.0.1
localhost has IPv6 address ::1
```

Mit der Option `-v` zeigt `host` wesentlich mehr Informationen an.

5.1.8 Links

(Linux) Netzwerk Konfiguration

<http://dozent.maruweb.de/material/netzwerkclient.shtml>

Debian Reference – Netzwerk-Konfiguration

<http://www.debian.org/doc/manuals/debian-reference/ch-gateway.de.html>

Manitu Wiki – Kategorie Linux (Server) – Sysctl-Variablen

[http://wiki.manitu.de/index.php/Server:Sysctl-Variablen_\(sysctl.conf\)](http://wiki.manitu.de/index.php/Server:Sysctl-Variablen_(sysctl.conf))

Ubuntuusers Wiki – Kategorie Interfaces (Datei)

<http://wiki.ubuntuusers.de/interfaces>

⁹Alternativen sind `dig` und `nslookup`