

**ZSL**

Zentrum für Schulqualität  
und Lehrerbildung  
Baden-Württemberg

# LINUX ESSENTIALS

## 14: Den Computer im Netzwerk einbinden



**Tobias Heine**

[tobias.heine@springer-schule.de](mailto:tobias.heine@springer-schule.de)

**Andreas Mundt**

[a.mundt@lehrerfortbildung-bw.de](mailto:a.mundt@lehrerfortbildung-bw.de)

**Jan Nathan**

[jan.nathan@lehrerfortbildung-bw.de](mailto:jan.nathan@lehrerfortbildung-bw.de)

## 4.4 Your Computer on the Network (weight: 2)

<b>Weight</b>	2
<b>Description</b>	Querying vital networking configuration and determining the basic requirements for a computer on a Local Area Network (LAN).

### Key Knowledge Areas:

- Internet, network, routers
- Querying DNS client configuration
- Querying network configuration

### The following is a partial list of the used files, terms and utilities:

- route, ip route show
- ifconfig, ip addr show
- netstat, ss
- /etc/resolv.conf, /etc/hosts
- IPv4, IPv6
- ping
- host

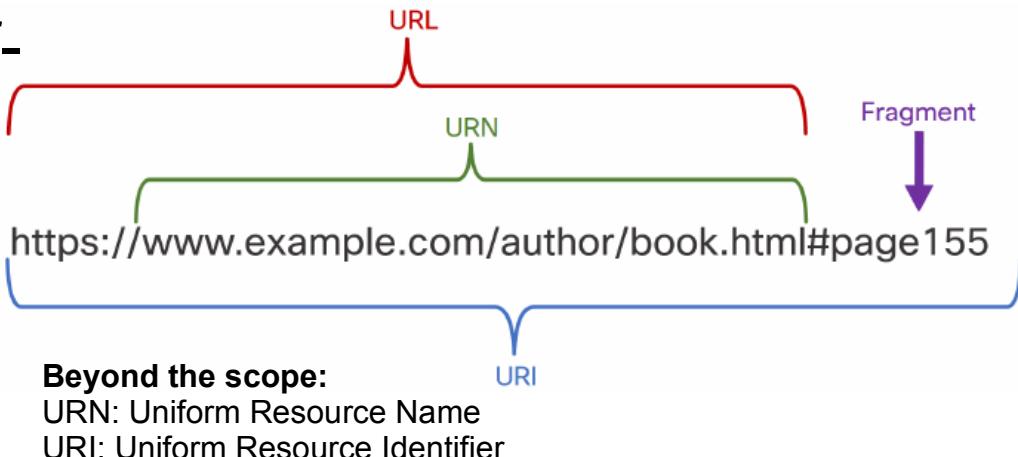
Viel Infos zu einem hochinteressanten Thema:

- **Teil 1: Begriffe**
  - Wichtig für die Prüfung!
- **Teil 2: IP-Adressierung**
  - Gewisse Tiefe aber didaktisch reduziert
  - Wichtig für Alltag/Unterricht/Verständnis
    - ... nur in Grundzügen für die Prüfung.
- **Teil 3: Dienste, Tools und Konfig-Dateien**
  - Wichtig für die Prüfung!

**Werbeblock:** Das Thema „Netzwerke“ ist absolut interessant und kommt oftmals zu kurz. Wer hier eine intensive Schulung nicht scheut, kann eine 6teilige Fortbildung (6 x 2,5 Tage) ins Auge fassen: **CCNA**. Diese wird jährlich angeboten.  
<https://lfbo.kultus-bw.de/lfb/termine/XMX8X> (Super Sache; Tut aber weh!)

- **Host:** Jedes Gerät, das mit anderen Gerät in einem Netzwerk kommuniziert. Diese haben in der Regel einen **Hostnamen**.
- **Netzwerk:** Mindestens zwei verbundene Hosts, die miteinander kommunizieren können.
- **Internet:** ein öffentlich zugängliches Netzwerk, das Millionen von Hosts verbindet.
- **Server:** Ein Host der einen oder mehrere Dienste im Netzwerk anbietet (z. B. Webserver, Mailserver, Druckerserver, etc.). 
- **Client:** Ein Endgerät (Host) am Arbeitsplatz. In der Regel ein PC oder Notebook. Aber auch ein Netzwerk-Drucker. 
- **Switch:** Ein Netzwerkgerät, das 4, 8, 16, 24, 48 oder mehr Anschlüsse besitzt und dementsprechend viele Hosts miteinander verbinden kann. 
- **Router:** Ein Netzwerkgerät das Netzwerkverkehr auf Basis von IP-Adressen in die entsprechende Richtung leitet. 

- **TCP/IP: Transmission Control Protocol/Internet Protocol:** Sammlung von Protokollen die definieren wie die Kommunikation zwischen Hosts funktioniert. (Protokoll = Regeln)
- **Ethernet:** Der gebräuchlichste Weg, wie Hosts physikalisch verbunden werden. Umfassen Kabeltypen, Stecker, Übertragungsformen. Es werden verschiedene Übertragungsgeschwindigkeiten unterstützt: 10/100/1000/... Mbps
- **URL: Uniform Resource Locator:** (Web-Adresse) ... wird zum auffinden von Ressourcen im Netz verwendet.



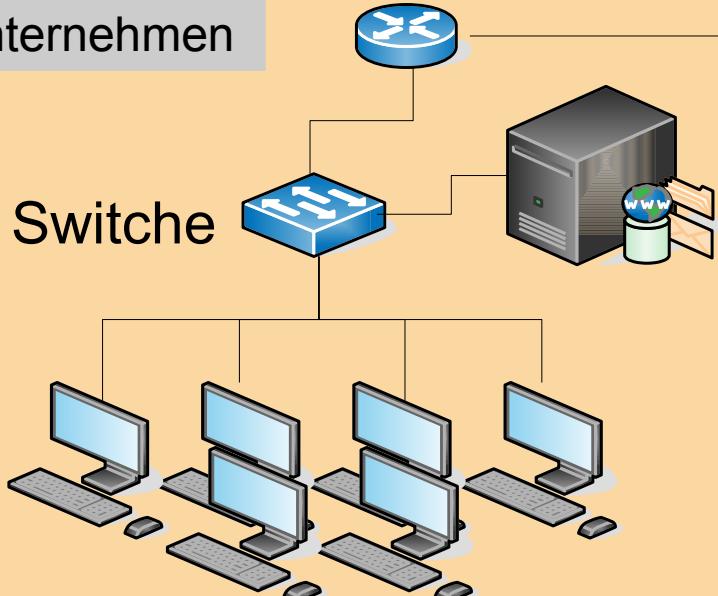
# Geräte im Netzwerk (vereinfachter Überblick)

## Domain Name System (DNS)

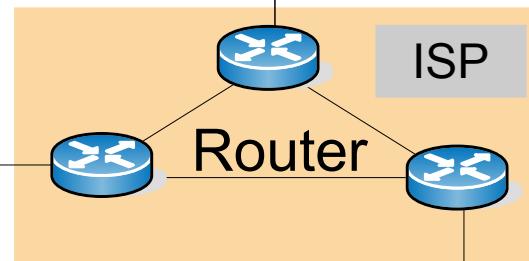


Host (Server:  
Web, Drucker,  
Mail, etc.)

Unternehmen



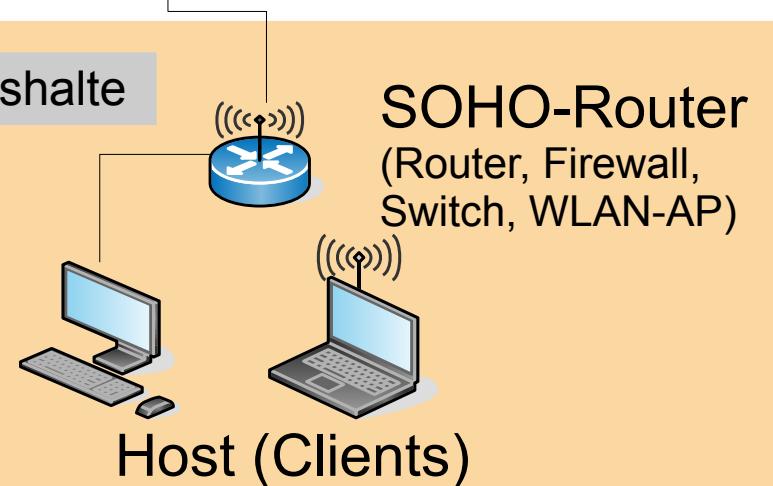
Switches



ISP

Router

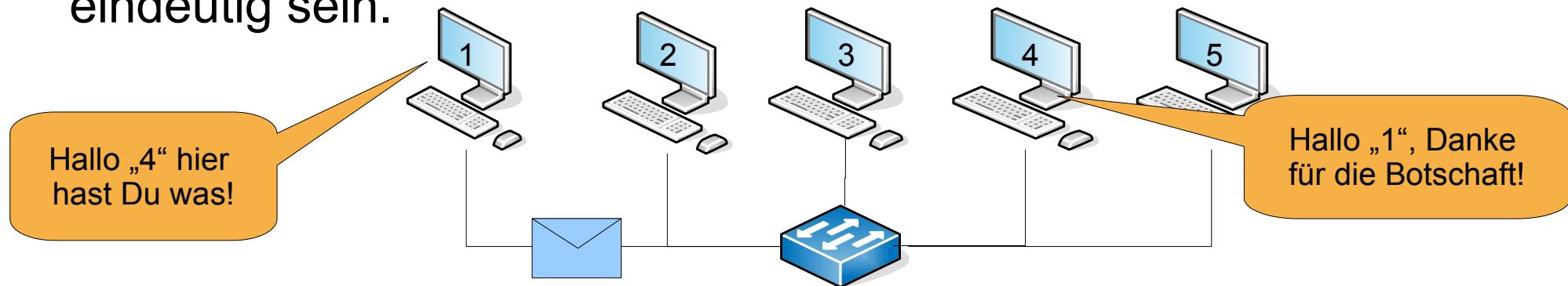
Haushalte



SOHO-Router  
(Router, Firewall,  
Switch, WLAN-AP)

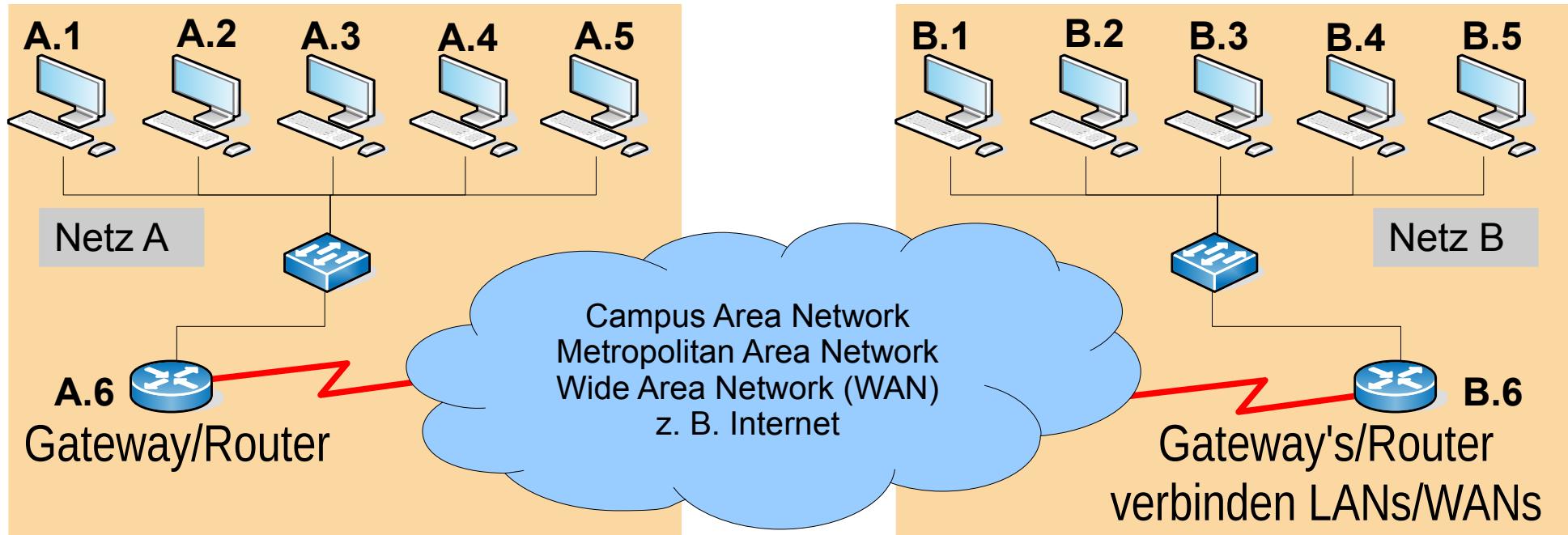
Host (Clients)

- Damit Rechner untereinander kommunizieren können, benötigen sie im LAN eine Rechner-Adresse (vgl. Telefonnummer).
- Um zielgerichtet kommunizieren zu können, muss diese Adresse eindeutig sein.



- Nur der angesprochene Rechner reagiert. Alle anderen Rechner ignorieren die Nachricht (oder bekommen diese nicht)
- Infos gehen (vereinfacht) über **IP-Pakete** über das Netz.
- Diese enthalten Ziel-Adresse, Absender-Adresse und Daten

# Adressen bei mehreren Netzen (immer noch vereinf.)



- Mehrdeutige Adressen werden durch eine Netzwerk-Identity / Netzwerk-Prefix verhindert
- Informationen liegen in

- Verwenden eine 32-Bit-Rechneradresse
- Rechneradresse enthält Netz- und Rechner-ID

**Vergleich:** Gegeben sind folgende Telefonnummern. Wo endet die Vorwahl und wo beginnt die Durchwahl?

0621615301

0622112228

Mannheim: 0621  
Heidelberg: 06221

**110000001010100000000000100001010**

Netz-ID

Rechner-ID

Trennlinie für beide ID's wird durch eine überlagerte 32-Bit Netz-Maske realisiert u. angegeben:

**111111111111111111111111100000000**

- 32-Bit-Ziffern sind schlecht zu merken.
  - Zur einfacheren Darstellung werden die einzelnen Bytes (4x8 Bit) deshalb getrennt ...
  - ... und als Dezimalzahl geschrieben.
  - Zur Trennung wird jeweils ein Dezimalpunkt verwendet.

11000000	10101000	00000001	00001010
----------	----------	----------	----------

192 . 168 . 1 . 10

- Damit ergibt sich für dieses Beispiel die IPv4-Adresse „192.168.1.10“.

- Netzmaske → wie bei den IPv4-Adressen die Variante "dotted-decimal"/"dotted-quad" um eine übersichtliche Darstellung zu erhalten.
  - Im vorigen Beispiel:  
**1111111111111111111111111100000000**  
→ **255.255.255.0**
  - Gesamt: 192.168.1.10/255.255.255.0
- Die einfache Struktur einer Netzmaske erlaubt eine noch einfachere Art der Darstellung
  - nur die Anzahl der „1“-Bit's angeben
  - Beispiel: 192.168.1.10/**24**
  - Wird auch als CIDR-Darstellung bezeichnet.

# Wie viele Hosts gibt es im Netzwerk

- Die Netzmaske lässt Rückschlüsse auf die Größe eines Netzwerkes zu (Potential)

Netzmaske des Beispiels:

11111111111111111111111111111111	00000000
255 . 255 . 255 .	0

IP-Adresse des Beispiels:

110000001010100000000001	00001010
192 . 168 . 1 .	10

Host-ID: Die IP-Adresse kann in diesem Bereich, jegliche Kombination annehmen.

**2<sup>8</sup> Adressen = 256 IP-Adressen  
in diesem Beispielnetzwerk**

- Unterste Adresse eines Netz-Adressbereichs
  - Netz-Identity (Netz-ID)
  - wird nicht für einen Rechner verwendet
  - **im Beispiel: 192.168.1.0**
- Die oberste Adresse eines Netz-Bereichs
  - Broadcast-Adresse
  - Pakete erreichen alle Rechner in einem LAN
  - **im Beispiel: 192.168.1.255**
- In unserem Netzwerk sind damit  
 **$2^8 - 2$  IP-Adressen verwendbar = 254**

- Ein **größeres** Netz könnte ein Rechner mit folgender IP-Adress-Information haben:

172.16.17.10/16

IP-Adressen gesamt:  $2^{16} = 65536$

Netz-IP: 172.16.0.0/16

Broadcast-IP: 172.16.255.255/16

**verwendbare IP-Adressen:  $2^{16} - 2 = 65534$**

- Ein **kleineres** Netz könnte ein Rechner mit folgender IP-Adress-Information haben:

192.168.8.2/28

IP-Adressen gesamt:  $2^4 = 16$

Netz-IP: 192.168.8.0/28

Broadcast-IP: 192.168.8.15/28

**verwendbare IP-Adressen:  $2^4 - 2 = 14$**

- Problem von IPv4: IP-Adressen gehen zu Neige!  
IPv4-Adresse ist 32 Bit groß  
 $2^{32} = 4.294.967.296$   
“Milchmännchenrechnung“: nicht mal jeder Erdenbürger könnte eine eigene IP-Adresse haben.
- Ein Lösungsansatz: **NAT (Network Adress Translation)**:
  - Haushalt bekommt **eine eindeutige öffentliche IP-Adresse**.
  - Diese Adresse liegt auf der Internetschnittstelle des Routers.
  - Alle Rechner im lokalen Netz erhalten eine eigene **private Adresse**.
  - Sobald der Verkehr den Router passiert, wird die private Absender-Adresse durch die öffentliche Absender-Adresse ersetzt.
  - Der Router führt eine Liste darüber, welcher Client, welche Anfrage führt
- Öffentliche Adresse anzeigen: <https://meineipadresse.de/>

- Für die Nutzung innerhalb eines LANs werden daher „private Adressen“ verwendet!
- Private Adressen werden im Internet **NICHT** geroutet.
- Es gibt drei Adressräume für Private Adressen, die je nach Größe des Netzwerks eingesetzt werden:
  - **10.0.0.0/8** (Adressen von 10.0.0.0 bis 10.255.255.255)  
 $2^{24} - 2 = 16.777.214$  IP-Adressen
  - **172.16.0.0/12** (Adressen von 172.16.0.0 bis 172.31.255.255)  
 $2^{20} - 2 = 1.048.574$  IP-Adressen
  - **192.168.0.0/16** (Adressen von 192.168.0.0 bis 192.168.255.255)  
 $2^{16} - 2 = 65.534$  IP-Adressen

- 128 Bit Größe → Schreibweise in Hex-Ziffern (= 340 Sextillionen Adressen → 340 + 36 Nullen)
- Acht Blöcke zu je 16 Bit → 4 Hex-Ziffern
- Trennzeichen → je ein Doppelpunkt
- Grundsätzlich:  
64-Bit-Prefix + 64-Bit Hostanteil (ohne Subnetting)
- Führende Nullen pro Block können weg gelassen werden.  
Extremfall :0000: → :0:
- Ein oder mehrere Nullblöcke ( ...:0:0:0:....) dürfen mit :: abgekürzt werden – **ABER NUR EINMAL PRO IPv6-ADRESSE** – da sonst mehrdeutig.

*your IPv4 internet fits  
4.294.967.296 times in my  
homenet*

- Beispiele zur kürzen Darstellung von IPv6-Adressen:

- 2003:**0000**:130F:**0000**:**0000**:09C0:876A:130B  
ist identisch mit 2003:**0**:130f:**0**:**0**:9c0:876a:130b  
**NICHT ERLAUBT:** ~~2003::130f::9c0:876a:130b~~
- FE80:0:0:0:0:0:1 → FE80::1
- 0:0:0:0:0:0:1 → ::1

Das einmalig auftretende :: ist so zu expandieren, dass am Ende wieder 128-Bit vorhanden sind.

- Ein Host, der IPv6-Adressen nutzt hat häufig zwei Adressen:

Link-Local: FE80::201/64 ← nur lokal gültig

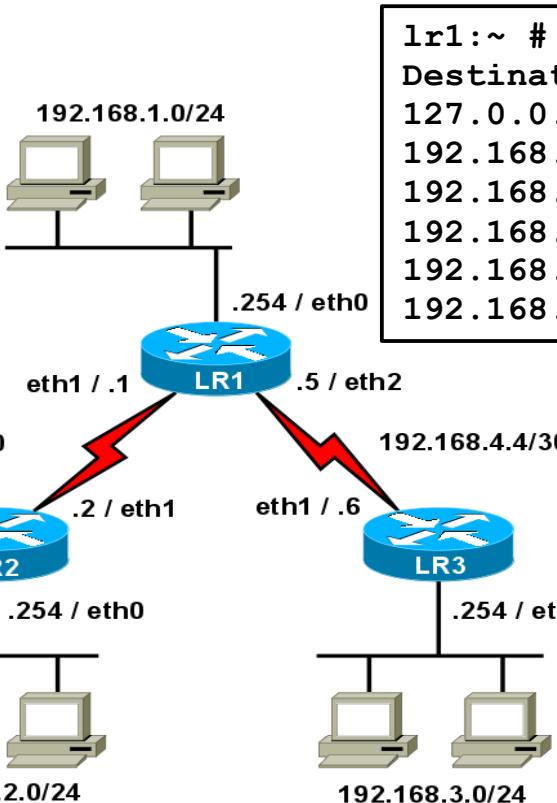
Global-Unicast: 2003::201/64 ← öffentliche Adresse

- Arbeit im lokalen Netz möglich, sobald Host eine **IPv4- und/oder IPv6-Adresse** besitzt.
- Netzwerk-Verkehr über das Internet läuft über Computer, die eine besondere Aufgabe haben: **Router**.
- Router kennen den Weg zu den einzelnen Netzen, durch Einträge in Ihrer **Routing-Tabelle**.
- „Normale“ Hosts (Server, Client-PCs, Netzwerkdrucker), kennen den Weg zu Ihrem nächsten Router, den sie als **(Default-)Gateway** nutzen.
- Router haben häufig ebenfalls einen Default-Gateway-Eintrag

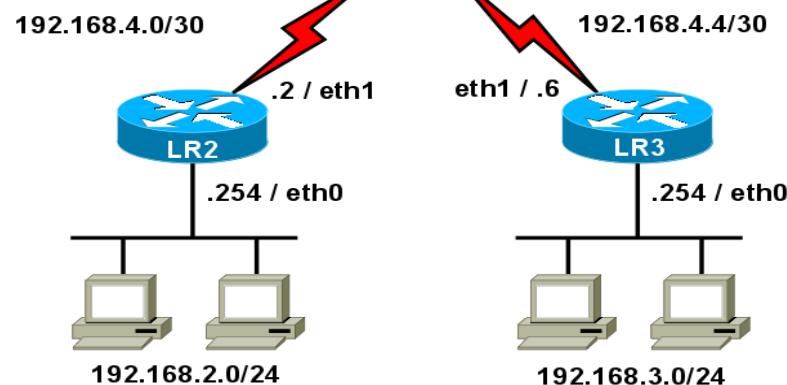


**Alle Richtungen**

# Ein viel zu komplexes Beispiel – gut für Verständnis



lr1:~ # route -n							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
192.168.2.0	192.168.4.2	255.255.255.0	UG	0	0	0	eth1
192.168.3.0	192.168.4.6	255.255.255.0	UG	0	0	0	eth2
192.168.4.0	0.0.0.0	255.255.255.252	U	0	0	0	eth1
192.168.4.4	0.0.0.0	255.255.255.252	U	0	0	0	eth2

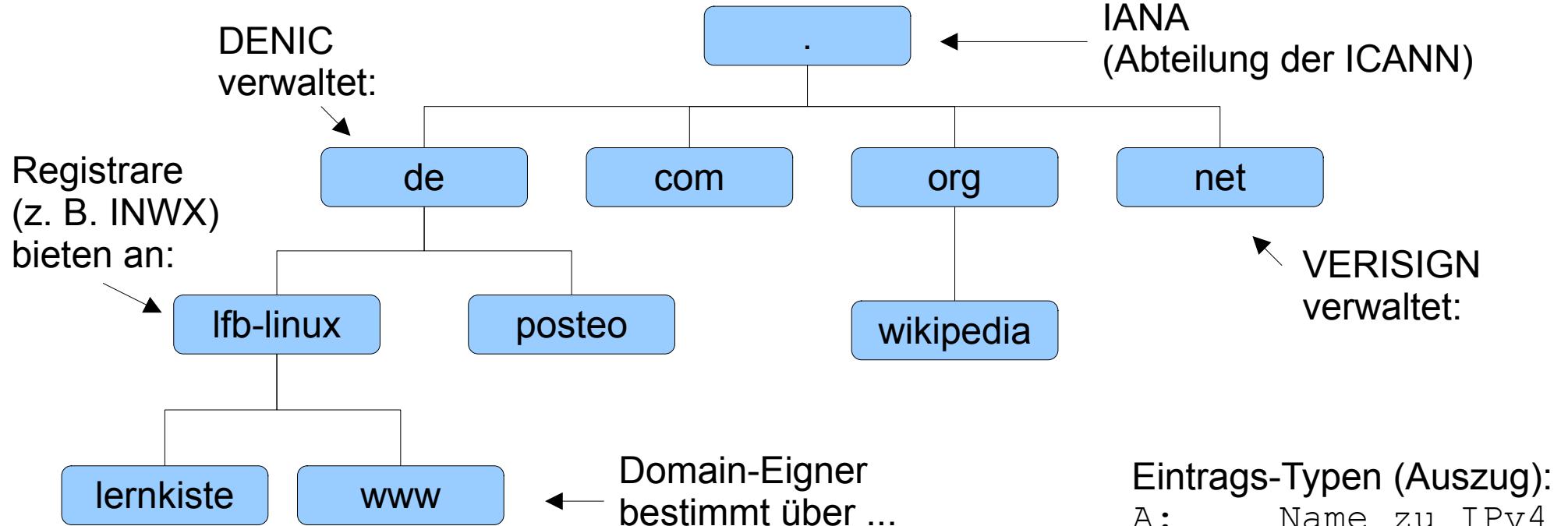


lr3:~ # route -n				
Destination	Gateway	Genmask	Flags	Iface
127.0.0.0	0.0.0.0	255.0.0.0	U	lo
192.168.1.0	192.168.4.5	255.255.255.0	UG	eth1
192.168.2.0	192.168.4.5	255.255.255.0	UG	eth1
192.168.3.0	0.0.0.0	255.255.255.0	U	eth0
192.168.4.0	192.168.4.5	255.255.255.252	UG	eth1
192.168.4.4	0.0.0.0	255.255.255.252	U	eth1

lr2:~ # route -n							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.4.1	0.0.0.0	UG	0	0	0	eth1
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
192.168.1.0	192.168.4.1	255.255.255.0	UG	0	0	0	eth1
192.168.2.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

- Kommunikation mit IP-Adressen somit vollständig.
- Webseitenaufruf geht über die IP-Adresse: z. B.  
<http://193.99.144.80>\*
- Menschen brauchen ein System, um das Ziel mit einem Namen ansprechen: z.B.  
<http://heise.de>\*
- Wird über das **Domain Name System** realisiert. Ein hierarchisches System zu Verwaltung von Zuordnungen (Namen zu IP-Adressen u. v. m.) → „Telefonbuch des Internets“

## Hierarchischer Aufbau:



### Mal einen Blick reinwerfen:

<https://root-servers.org/>

<https://www.icann.org/en/accredited-registrars>

<https://www.denic.de/>

### Eintrags-Typen (Auszug):

A:	Name zu IPv4
AAAA:	Name zu IPv6
MX:	Mail Exchange
PTR:	IP zu Name
TXT:	Text
NS:	Nameserver

## Zusammenfassend ...

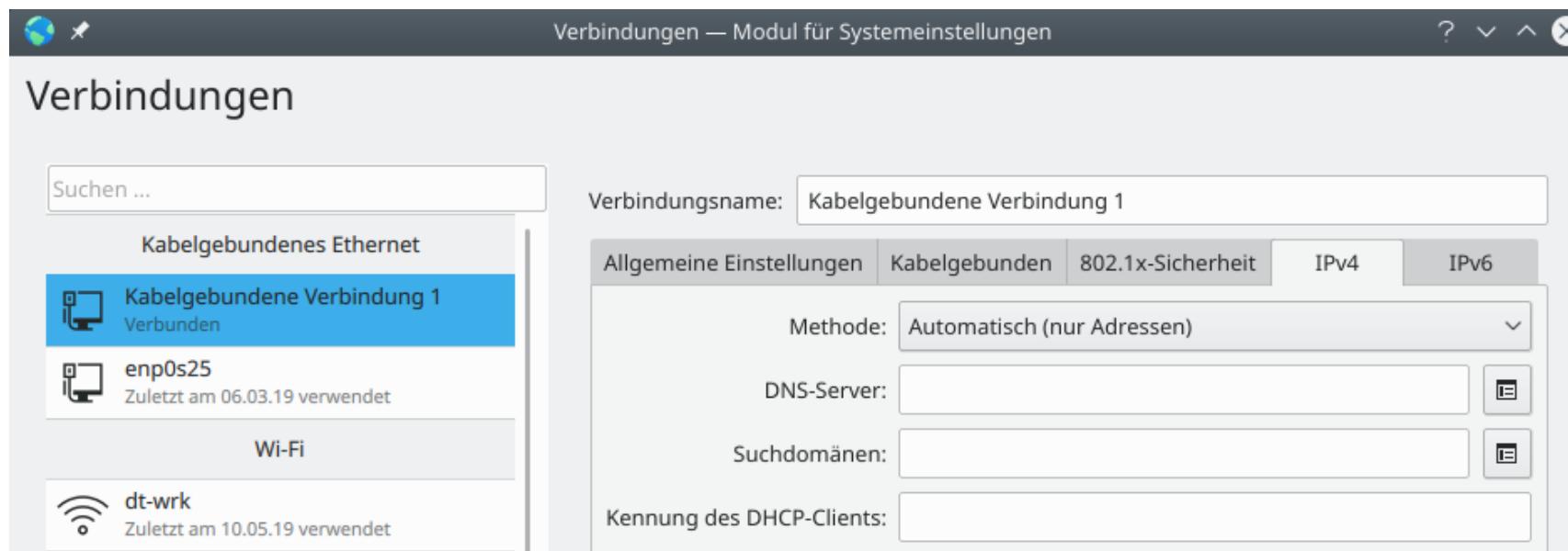
Damit ein Rechner zielgerichtet über das Internet kommunizieren kann, wird folgende Konfiguration benötigt:

- **IPv4-Adresse** und Subnetzmaske oder 192.168.1.0 mit 255.255.255.0 oder /24
- **IPv6-Adresse** mit gültigem Präfix 2003:db8:acad::1
- Ein Eintrag für das **Default-Gateway** (der nächste Router)
- Ein (besser zwei) Einträge für die zu nutzenden **DNS-Server**

# Netzwerk-Geräte konfigurieren

Zwei grundlegende Fragen bei der Implementierung:

- Kabelgebunden oder Kabellos (wired or wireless)
- IP-Adresse **manuell** vergeben oder **automatisch** über das Protokoll **DHCP (Dynamic Host Konfiguration Protocol)**



## Interface mit IP-Adress-Informationen versehen

- /etc/network/interfaces
  - Interface Debian und Ubuntu < 18.04
- /etc/netplan/<Name>.yaml
  - Interface Ubuntu >= 18.04
- /etc/sysconfig/network-scripts/ifcfg...
  - RHEL, CentOS, ...: pro Interface eine Datei
- /etc/sysconfig/network-scripts/ifcfg...
  - SLES, openSUSE: pro Interface eine Datei
- Nach Veränderung von Netzwerk-Konfigurationsdateien empfiehlt sich ein Neustart des Netzwerks:  
`sudo systemctl restart networking.service`

## Neben dem Interface für das Netzwerk wichtig:

- /etc/resolv.conf  
→ Beinhaltet die DNS-Server die der Host nutzt

```
[19:45] srv: ~# cat /etc/resolv.conf
domain home.tbshn.de
search home.tbshn.de
nameserver 192.168.180.8
nameserver 192.168.180.11
```

- /etc/hosts  
→ lokale Zuweisung von Hostnamen zu IP-Adressen
- /etc/nsswitch.conf  
→ In welcher Reihenfolge werden Namens-IP-Zuweisungen geprüft.

- IP-Adresse auslesen/modifizieren: `ifconfig` oder `ip addr`
- Routing-Tabelle auslesen/modifizieren: `route` oder `ip route`
- Analyse des Netzwerks: `netstat`
- Gegenstelle erreichbar: `ping <Adresse>`
- Pfad zur Gegenstelle: `traceroute <Adresse>`
- DNS-Namen auflösen: `dig` und `host`
- Sicherer Zugriff auf Remote-Host: `ssh`
- Sicheres Kopieren zwischen lokalem + Remote-Host: `scp`

Sind net-tools installiert? → `apt install net-tools`

# Interface-Adressen ändern/auslesen (Auszug)

- Altes Tool: `ifconfig` (auslesen nur mit root-Rechten)

```
ifconfig enp0s31f6
```

```
ifconfig enp0s31f6 192.168.190.199 netmask 255.255.255.0
```

```
ifconfig enp0s31f6 down
```

```
ifconfig enp0s31f6 hw ether 08:6D:41:01:02:03
```

```
ifconfig enp0s31f6 up
```

← Ha ... ich bin  
jetzt ein  
Apple-Gerät

- Neues Super-Tool: `ip` und dessen Objekt `address`

```
ip address
```

```
ip address show enp0s31f6 → abkürzen mgl.: ip addr s enp0s31f6
```

```
ip address add 192.168.190.198/24 dev enp0s31f6
```

- Altes Tool `route` und `netstat -r` führt zu dieser Ausgabe:

Kernel-IP-Routentabelle

Ziel	Router	Genmask	Flags	MSS	Fenster	irtt	Iface
default	192.168.190.254	0.0.0.0	UG	0	0	0	ens18
192.168.190.0	0.0.0.0	255.255.255.0	U	0	0	0	ens18

- Neues Super-Tool `ip` mit Objekt `route` führt zu dieser Ausgabe:

```
[20:15] srv: ~# ip route
default via 192.168.190.254 dev ens18 proto kernel onlink
192.168.190.0/24 dev ens18 proto kernel scope link src 192.168.190.1
```

- Auch Route hinzufügen möglich ... Beispiel mit `route`:

```
route add default gw 192.168.190.254
```

```
route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.190.253
```

- `netstat` → Netzwerkverbindungen auf dem Host anzeigen:
  - `netstat -r` → Gleiche Ausgabe wie `route`
  - `netstat -ei` → Gleiche Ausgabe wie `ifconfig`
  - `netstat -n` → Namen nicht auflösen
  - `netstat -tulpn` → Ein Ansatzpunkt Server-Troubleshooting (nur TCP, UDP, im Listening, ohne PID und ohne Hostnamen)
- `ss` (socket statistics) → Alternative zu netstat (Neuer, besser, mehr Features ... und ... ich nutz es nicht ;-))
  - `ss -tulpn` geht auch
  - `ss -s` → Summary

- Mit `ping` ein Ziel prüfen (ist es erreichbar):

```
heine@lernkiste:~$ ping -c 1 9.9.9.9
PING 9.9.9.9 (9.9.9.9) 56(84) bytes of data.
64 bytes from 9.9.9.9: icmp_seq=1 ttl=57 time=11.2 ms
```

Läuft normal unendlich durch (-c 1 setzt nur EINEN Ping)

Rückschlüsse auf Erreichbarkeit, Geschwindigkeit und Entfernung in Hops (passierende Router)

- Details zu passierenden Routern mit `traceroute`:

```
heine@lernkiste:~$ traceroute -n 9.9.9.9
traceroute to 9.9.9.9 (9.9.9.9), 30 hops max, 60 byte packets
 1  138.201.164.17  0.103 ms  0.078 ms  0.064 ms
 2  100.91.33.129  0.445 ms  0.433 ms  0.429 ms
 3  213.239.229.217  0.555 ms  213.239.229.221  0.724 ms  213.239.229.217  0.504 ms
 4  213.239.252.249  2.663 ms  213.239.252.245  4.624 ms  213.239.252.249  2.632 ms
 5  * * *
 6  9.9.9.9  11.240 ms !X  11.041 ms !X  11.074 ms !X
```

- DNS-System mit `dig` abfragen:

```
dig lfb-linux.de → Abfrage A-Record (IPv4)
```

```
dig -x 185.181.104.242 → Reverse Lookup
```

```
dig -t AAAA lfb-linux.de → Abfrage AAAA-Record (IPv6)
```

```
dig -t txt lfb-linux.de → Abfrage TXT-Record
```

- host ist ein „einfacheres“ Tool zur DNS-Abfrage. Ähnlich:

```
host lfb-linux.de
```

```
host 185.181.104.243
```

```
host -t CNAME lfb-linux.de
```

```
[21:06] srv: ~# host -t CNAME scb.home.tbshn.de
scb.home.tbshn.de is an alias for web.home.tbshn.de.
```

## Secure Shell via SSH

- Verbindungsaufbau:

```
ssh heine@lernkiste.lfb-linux.de
```

Wenn „yes“ dann landet der Public-SSH-Host-Key des Servers in der  
~/.ssh/known\_hosts.

- Bei Erst-Verbindung prüfen ob der angezeigte Fingerprint ...

```
L$ ssh heine@lernkiste.lfb-linux.de
The authenticity of host 'lernkiste.lfb-linux.de (138.201.164.20)' can't be established.
ECDSA key fingerprint is SHA256:VjkphzfCQ06deHrgcqDWolx7yvJV5LPBZcF4S3YFS2s.
Are you sure you want to continue connecting (yes/no/[fingerprint])? [
```

- ... einem vorhanden Fingerprint des Servers entspricht:

```
heine@lernkiste:~$ for i in /etc/ssh/*.pub; do ssh-keygen -lf $i; done
1024 SHA256:+gzi+NR5+pks/1PzBBHHBeBGda7W8flLIqkVVhKfbZQ root@lernkiste (DSA)
256 SHA256:VjkphzfCQ06deHrgcqDWolx7yvJV5LPBZcF4S3YFS2s root@lernkiste (ECDSA)
256 SHA256:1ee5NJnLm0dul0ujIVw3woFw/P8rA3uc11wn/RItXkI root@lernkiste (ED25519)
3072 SHA256:awcvzB53sV5BFuy4TPzWQe+qP43F0BZqz7W0ZCkVfTI root@lernkiste (RSA)
```

- Mit dem Kommando `scp` (Secure Copy) kann man Daten von einem lokalen host mit einem Remote-Host hin- und her kopieren. Hier nicht relevant aber ein absolutes Basis-Tool.
- **Labor zur „Netzwerk-Simulation“** (siehe Moodle-Kurs)  
... leider schon steinalt. Gute Erfahrung im Unterricht für einen praktischen Einblick
- **Workbook:** IP-Adressing and Subnetting (siehe Moodle-Kurs) ... mit Rechenbeispielen für Subnetze und VLANs
- **Nochmal der Werbeblock:**  
Cisco CCNA-Zertifizierung  
(für die Ambitionierten)  
... Danach bleiben keine Fragen  
für Standard-Netze offen!

Fragen ...

