

# IPv4-Laborübung

## Allgemeine Informationen zu IPv4

Im folgenden Abschnitt werden einige grundlegenden Befehle aufgeführt mit denen IPv6-Adressen angezeigt, gesetzt bzw. gelöscht werden können.

Weiterhin kann es notwendig sein ein Interface zu de-/aktivieren.

**HINWEIS:** Bei administrativen Eingriffe (de-/aktivieren, setzen, löschen), muss vor diesen Befehlen `sudo` vorangestellt werden. Es gibt prinziell die beiden Befehle **ifconfig** und den neueren (mächtigeren) **ip**. Im Folgenden werden die wichtigsten Befehle erläutert.

## Anzeige der aktuellen IPv4-Adressen

Mit dem folgenden Befehlen werden die IPv4-Adressen auf dem System ausgegeben (Hinweis: Wird -4 nicht angegeben, dann werden auch die IPv6-Adressen angezeigt.):

Befehl: **ip -4 addr**

Ausgabe:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    inet 192.168.16.101/24 brd 192.168.0.255 scope global eth0
        valid_lft forever preferred_lft forever
```

Befehl: **ifconfig**

Ausgabe:

```
eth0      Link encap:Ethernet  Hardware Adresse 00:26:18:bb:a7:71
          inet Adresse:192.168.0.10  Bcast:192.168.0.255
Maske:255.255.255.0
          inet6-Adresse: fe80::226:18ff:febb:a771/64
Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metrik:1
          RX-Pakete:104774 Fehler:331 Verloren:0 Überläufe:331 Fenster:0
          TX-Pakete:67555 Fehler:0 Verloren:0 Überläufe:0 Träger:0
          Kollisionen:0 Sendewarteschlangenlänge:1000
          RX-Bytes:126793873 (126.7 MB)  TX-Bytes:21992168 (21.9 MB)
```

## Aktivieren/Deaktivieren eines Interfaces

Mit dem folgenden Befehlen werden Interfaces de-/aktiviert. Durch das Deaktivieren einer

Schnittstelle, wird auch deren temporäre Konfiguration gelöscht. Durch anschließendes Aktivieren, wird die Default-Konfiguration aus der Datei `/etc/network/interfaces` geladen.

Befehl:

```
ip link set dev <interface> up  
ip link set dev <interface> down
```

Befehl:

```
ifconfig <interface> up  
ifconfig <interface> down
```

Alternativ dazu kann ein Interface auch komplett zurückgesetzt werden: ACHTUNG: Danach muss ALLES neu konfiguriert werden, also Vorsicht!

Befehl: **ip addr flush eth0**

## Setzen einer neuen IPv4-Adresse

Mit dem folgenden Befehlen werden IPv4-Adressen gesetzt.

Befehl: **ip addr add <ipv4address>/<prefixlength> dev <interface>**

Befehl: **ifconfig <interface> <ipv4address>/<prefixlength>**

Beispiel: `ifconfig eth0 10.0.0.1/8`

## Löschen einer IPv4-Adresse

Mit dem folgenden Befehlen werden IPv4-Adressen gelöscht.

Befehl: **ip addr del <ipv4address>/<prefixlength> dev <interface>**

Befehl: **ifconfig <interface> del <ipv4address>/<prefixlength>**

HINWEIS: Sollte mit `ip` eine weitere IP-Adresse hinzugefügt worden sein, dann kann diese auch nur mit dem entsprechenden `ip`-Befehl wieder gelöscht werden.

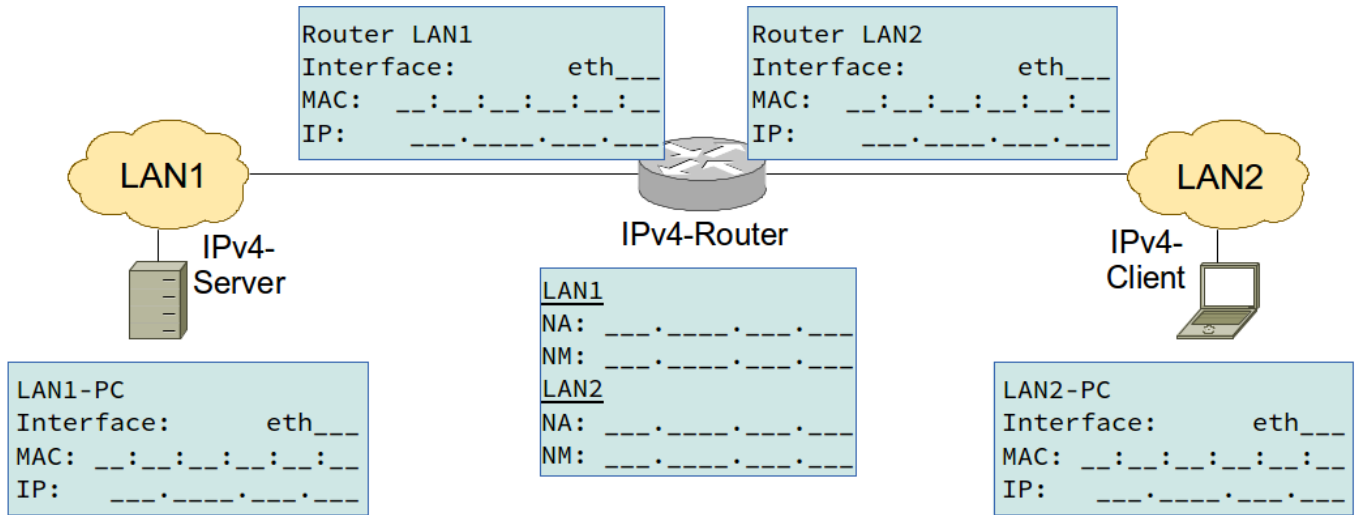
## Erreichbarkeit überprüfen (ping)

Mit dem `ping`-Befehl kann die Erreichbarkeit eines entfernten Systems überprüft werden.

Befehl: **ping <ipv4address></code>**

## Das Laborszenario (Aufgaben)

Das System besteht aus drei Maschinen. Einem **Client** einem **Router** und einem **Server**.<sup>1)</sup>



### Planung

1. Planen Sie Ihr Netz. Wählen Sie für das LAN1 und LAN2 jeweils unterschiedliche Subnetze.

Tragen Sie Ihre gewählten Daten in die folgende Tabelle:

| LAN | Netz  | Netzprefix |
|-----|-------|------------|
| 1   | _____ | _____      |
| 2   | _____ | _____      |

| Gerät   | Schnittstelle | IP-Adresse | MAC-Adresse | Default-Gateway |
|---------|---------------|------------|-------------|-----------------|
| Router1 | eth_          | _____      | _____       | _____           |
| Router2 | eth_          | _____      | _____       | _____           |
| Server  | eth_          | _____      | _____       | _____           |
| Client  | eth_          | _____      | _____       | _____           |

### Aufgaben

1. Notieren Sie sich alle relevanten Daten (Schnittstellen, MAC, IP,...) (s. Skizze)
2. Verwenden Sie die Befehle am Anfang dieser Seite, um die gewünschte Konfiguration herzustellen.
3. Zeichnen Sie die Kommunikation folgende Kommunikation mit wireshark auf. Verfolgen Sie den ping und achten Sie dabei vor allem auf die sich ändernden MAC-Adressen
4. Führen Sie einen Ping jeweils zwischen dem Server/Router und Client/Router aus.
5. Führen Sie einen Ping zwischen Server und Client aus.
6. Überprüfen Sie die gelernten IP/MAC Kombinationen mit dem Befehl **arp -a**

Auf der Router muss das sogenannte forwarding aktiviert werden. Hierdurch werden die beiden Schnittstellen nicht getrennt betrachtet, sondern der Router leitet Pakete von einem Netz in das andere weiter (eben: forward). Dieser Vorgang wird als routing bezeichnet.

Hier wird erklärt wie das forwarding aktiviert wird.

Damit die beiden Clients das jeweils andere Netz erreichen können, müsse diese entweder per route-Befehl das Netz „lernen“ oder aber ein Default-Gateway<sup>2)</sup> kennen. In beiden Fällen dienen die jeweiligen IP-Adressen des Routers in den beiden Netzen als Gateway-Adresse.

An dieser **Stelle** wird erklärt wie die Standard-Route gesetzt wird.

## Verwendung von Skripten zur Konfiguration

**TIP:** Da Sie wahrscheinlich häufiger umkonfigurieren werden, bietet sich ein kurzes Shell-Skript <sup>3)</sup> an. Wie man ein solches Shell-Skript erstellt und nutzt ist [auf einer eigenen Seite](#) erklärt. Die folgenden Beispiel-Skripte sind auf die eigenen Bedürfnisse anzupassen.

Beispielskripte mit ip:

[ip\\_labor1.sh](#)

```
#!/bin/bash
# alle IP-Adressen löschen
sudo ip addr flush eth0

# IP-Adresse 192.168.16.<PLATZ+100> auf Interface eth0 setzen
sudo ip addr add eth0 192.168.16.<PLATZ+100>

# Interface eth0 aktivieren
sudo ip link set dev eth0 up
# Default-Route setzen (Standard-Gateway)
sudo ip route add default via 192.168.16.1
```

Durch die Verwendung von Variablen kann das Ganze etwas universeller eingesetzt werden. Mut zum Experiment!

[ip\\_labor2.sh](#)

```
#!/bin/bash
IF1=eth1
IP_ROUTER1=10.0.0.1/8 # Variable für die Konfiguration der IP-Adresse
des Routers
sudo ip addr flush $IF1
sudo ip addr add $IP_ROUTER1 dev $IF1
sudo ip link set dev $IF1 up
ip -4 addr show $IF1
```

Beispielskripte mit ifconfig:

1)

Die Bezeichnungen sind vollkommen willkürlich und dienen nur der besseren Identifizierbarkeit.

2)

Standard-Gateway

3)

vgl. Batch-Datei unter Windows

From:

<https://www.kopfload.de/> - **kopfload - Lad Dein Hirn auf!**

Permanent link:

[https://www.kopfload.de/doku.php?id=lager:lok\\_netze:ipv4labor&rev=1400006063](https://www.kopfload.de/doku.php?id=lager:lok_netze:ipv4labor&rev=1400006063)

Last update: **2025/11/19 16:13**

