

# Aufwendiges statisches Routing Szenario-Laborübung (IPv4)

## Ziel

In dieser Laborübung soll mittels Kommandozeile <sup>1)</sup> eine Gruppe von Routern miteinander vernetzt werden. Die notwendigen Routen in die unterschiedlichen Netze sind dabei manuelle zu konfigurieren.

## Aufgabe

Die Router verwenden zwei Schnittstellen, wobei jeweils eine Schnittstelle in einem anderen Netz liegen. Die Router befinden sich paarweise in einem gemeinsamen Netz. Alle Netze sind so verschaltet, dass sie einen Kreis bilden. Netz soll mindestens ein Client konfiguriert werden, dass er einen der beiden Router als Default-Gateway nutzt. Das folgende Bild veranschaulicht den Aufbau.

## Durchführung

Folgenden Komponenten sind zu konfigurieren:

- Verkabelung: Ein Switch dient jeweils als ein „Netz“
- Router-Konfiguration: Die statischen Routen sind pro Router zu konfigurieren.
- Client-Konfiguration: Ein Default-Gateway ist pro Client zu konfigurieren.
- Testen mittels Ping: Die Verbindungen sind jeweils abschnittsweise zu prüfen.

## Allgemeine Informationen

In der [Laborübung zu IPv4](#) sind alle notwendigen Befehle zum Ändern von IP-Adressen beschrieben.

HINWEIS: Es wird dringend empfohlen die notwendigen Befehle per Shell-Script zu bearbeiten. Der grundsätzliche [Umgang mit Shell-Scripten](#) wird auf der folgenden Seite kurz erläutert.

## Router-Konfiguration

Beispiel Konfiguration für eine Router Schnittstelle. Das Script dient bewusst nur als grobes Gerüst. Es sind entsprechende Ergänzungen vorzunehmen.

[router\\_labor.sh](#)

```
#!/bin/bash
# alle IP-Adressen an eth1 löschen
sudo ip addr flush eth1
```

```
# IP-Adresse 192.168.99.<PLATZ+100> auf Interface eth1 setzen
sudo ip addr add dev eth1 192.168.99.<PLATZ+100>/<PREFIX>

# Interface eth1 aktivieren
sudo ip link set dev eth1 up

# Route in ein ZIELNETZ setzen über (via) ein Gateway im eigenen Netz
über die Schnittstelle (dev) setzen.
# (nur auf Routern)
sudo ip route add <ZIELNETZ> via <IP_des_Gateways> dev eth1
```

### Aufbau des ip route-Befehls

Befehl/Parameter	Bedeutung	Beispiel
sudo	Administrativer Eingriff	sudo ip addr add...
ip	ip-Befehl zur Konfiguration des IP-Protokoll-Stacks inkl. Routing	sudo ip route add ...
route	Der Parameter route gibt den Unterbereich vom ip-Befehl an, der für die Routing-Tabellen zuständig ist	sudo ip route add ...
addr	Der Parameter addr gibt den Unterbereich vom ip-Befehl an, der für die IP-Adressen zuständig ist	sudo ip addr add ...
add/del	Hinzufügen/Löschen einer Komponente (IP-Adresse oder Route). Dem Parameter folgt die IP-Adresse (addr) bzw. das Zielnetz (route).	sudo ip addr add ...
via	Nur im Kontext route: Gibt das Gateway bzw. den Next Hop als IP-Adresse an.	sudo ip route add 10.0.0.0 via 10.0.0.2 ...
default	Standard-Gateway für alle unbekanntenen Netze	sudo ip route add default via 10.0.0.2 ...
dev	Mit dev wird das <b>Device</b> als die Schnittstelle angegeben. Z.B. eth0	dev eth0

## Client-Konfiguration

Beispiel Konfiguration für eine Client-Schnittstelle. Das Script dient bewusst nur als grobes Gerüst. Es sind entsprechende Ergänzungen vorzunehmen. Hier wird der ip route-Befehl mit dem Parameter default benutzt. Damit wird das Standard<sup>2)</sup>-Gateway gesetzt, also der Router, der die Pakete des Clients in fremde Netze bringt.

### client\_labor.sh

```
#!/bin/bash
# alle IP-Adressen an eth1 löschen
sudo ip addr flush eth1
```

```
# IP-Adresse 192.168.99.<PLATZ+100> auf Interface eth1 setzen
sudo ip addr add dev eth1 192.168.99.<PLATZ+100>/<PREFIX>

# Interface eth1 aktivieren
sudo ip link set dev eth1 up

# Default-Route setzen (Standard-Gateway)
sudo ip route add default via 192.168.99.<IP_des_Gateways> dev eth1
```

## Testen mittels Ping

**t.b.d.**

1)

Shell

2)

default

From:

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

Permanent link:

[https://www.kopfload.de/doku.php?id=lager:lok\\_netze:ipv4staticroute&rev=1504075579](https://www.kopfload.de/doku.php?id=lager:lok_netze:ipv4staticroute&rev=1504075579)

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

