

# IPv6-Laborübung

## Allgemeine Informationen zu IPv6

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 oder dauerhaft mit sudo -s in den root-Modus gewechselt werden!

**HINWEIS:** Da im zweiten Teil der Laborübung mit virtuellen Maschinen gearbeitet werden soll, müssen gewisse Grundeinrichtungen für die VMs vorgenommen werden. Die wichtigsten Punkte sind [hier](#) beschrieben.

## Ausgabe der aktuellen Adressen

Mit dem folgenden Befehlen werden die IPv6-Adressen auf dem System ausgegeben:

```
ip -6 addr
AUSGABE:
1: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 fe80::226:18ff:febb:a771/64 scope link
        valid_lft forever preferred_lft forever

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.

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

ODER:

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

## Setzen einer neuen IPv6-Adresse

Mit dem folgenden Befehlen werden IPv6-Adressen gesetzt.

```
ip -6 addr add <ipv6address>/<prefixlength> dev <interface>
```

ODER:

```
ifconfig <interface> inet6 add <ipv6address>/<prefixlength>
```

## Löschen einer IPv6-Adresse

Mit dem folgenden Befehlen werden IPv6-Adressen gelöscht.

```
ip -6 addr del <ipv6address>/<prefixlength> dev <interface>
```

ODER:

```
ifconfig <interface> inet6 del <ipv6address>/<prefixlength>
```

## Erreichbarkeit überprüfen (ping)

Da der ping-Befehl für IPv4-Adressen geschrieben wurde, muss ein andere Befehl für IPv6-Adressen genutzt werden. Um z.B. die Erreichbarkeit zur Adresse `fc00::1` zu überprüfen, kann folgender Befehl verwendet werden.

```
ping6 fc00::1
```

Falls das Ziel über mehrere Schnittstellen erreicht werden könnte (z.B. Router besitzt zwei Schnittstellen `eth0` und `eth1` mit je einer Link-Local-Adresse und das Ziel ist ebenfalls eine Link-Local-Adresse), dann muss über den Zonenindex (`%ZONENINDEX`) angegeben werden, welche ausgehende Schnittstelle (`eth0`) verwendet werden soll.

**Router:**

`eth0` → `fe80::a00:27ff:fe00:1/64`

`eth1` → `fe80::a00:27ff:fe00:2/64`

**Ziel:**

Client → `fe80::a00:27ff:fe2c:7e1e`

```
ping6 fe80::a00:27ff:fe2c:7e1e%eth0
```

Hier wird der Ping über Zonenindex „eth0“ des Routers an den Client geschickt.

## Das Laborszenario (Aufgaben)

Das System besteht aus drei Maschinen. Einem **Client** einem **Router** und einem **Server**.



Der Router übernimmt später die Funktion des Router Advertisements mittels `radvd`<sup>1)</sup> und die DHCPv6 Funktion mittels `isc-dhcp-server`.

Es wird von zwei Netzen ausgegangen. `fc00:1::/64` ist das lokale Netz in dem sich Amy befindet und `2001:db8:2::/64`<sup>2)</sup> ist das öffentliche Netz in dem Berny sich befindet. Der Router verbindet die beiden Netze.

Die ersten Laboraufgaben: [statische Routen](#) konfigurieren

1. Manuelle Konfiguration der Routen und IPv6-Adressen für ein **statisches Routing** ([Routing auf Router aktivieren](#))
2. Mit Wireshark die Kommunikation protokollieren

Die zweiten Laboraufgaben: [stateless Konfiguration](#)

1. Automatische Konfiguration **stateless (Router Advertisements)** mit `radvd`
2. Mit Wireshark die **Neighbor Discovery** protokollieren

Die dritten Laboraufgaben: [stateful Konfiguration](#)

1. Automatische Konfiguration **stateful (DHCPv6)** `isc-dhcp-server`
2. Mit Wireshark das DHCPv6-Protokoll mitschneiden

## Beispiele für statische Konfiguration

Im folgenden werden Beispiel-Konfigurationen aufgelistet, die entsprechend des eigenen Szenarios anzupassen sind.

### Client Lokal (Amy) und Öffentlich (Berny)

Die folgende Datei zeigt ein Beispiel für den lokalen Client (Amy). Der öffentliche Client ist sinngemäß im zweiten Netz zu konfigurieren.

[amy.sh](#)

```
#!/bin/bash
echo "#####"
echo "  Ausgangssituation AMY (Lokal)"
echo "#####"
```

```
echo "          IPv6-Adressen ALT"
ip -6 addr
echo "-----"
echo "          Routen ALT"
ip -6 route show
echo
echo "#####"
echo "  IPv6-Adressen Konfiguration und Gateway setzen"
echo "    setze fc00:1::100/64 auf eth0"
echo "    setze fc00:1::1/64 als Default GW"
echo "#####"
ip -6 addr add fc00:1::100/64 dev eth0
ip -6 route add default via fc00:1::1
echo "#####"
echo "          Neue Konfiguration"
echo "#####"
echo "          IPv6-Adressen !NEU!"
ip -6 addr
echo "-----"
echo "          Routen !NEU!"
ip -6 route show
```

## Router

Neben den beiden Schnittstellen ist auf dem Router zusätzlich die Routing-Funktion (Forwarding) zu aktivieren.

### router.sh

```
#!/bin/bash
echo "#####"
echo "          Ausgangssituation ROUTER"
echo "#####"
echo "          IPv6-Adressen ALT"
ip -6 addr
echo "-----"
echo "          Routen ALT"
ip -6 route show
echo
echo "#####"
echo "  IPv6-Adressen Konfiguration und Forwarding aktivieren"
echo "    setze fc00:1::100/64 auf eth0"
echo "    setze 2001:db8:2::1/32 auf eth1"
echo "    setze net.ipv6.conf.all.forwarding=1"
echo "#####"
ip -6 addr add fc00:1::1/64 dev eth0
ip -6 addr add 2001:db8:2::1/32 dev eth1
sysctl net.ipv6.conf.all.forwarding=1
```

```
echo "#####"  
echo "      Neue Konfiguration"  
echo "#####"  
echo "      IPv6-Adressen !NEU!"  
ip -6 addr  
echo "-----"  
echo "      Routen !NEU!"  
ip -6 route show
```

**HINWEIS:** Verwenden Sie für die IPv6-Netze ein Local-Unicast-Link-Netz und ein Global-Unicast-Netz bzw. eine Adresse aus dem für Dokumentation vorgesehenen Netzbereich (2001:db8::/32). Untersuchen Sie, ob der Client und der Server über die Link-Local-Adressen kommunizieren können (z.B. ping6).

## Der Router

### IPv6-Routing aktivieren

Für die erste Übung muss die VM, die als Router fungiert zunächst zum Router gemacht werden. Hierzu muss das IPv6-Forwarding aktiviert werden. Mit dem folgenden Befehl lässt sich das statische Routing aktivieren.

```
sudo sysctl net.ipv6.conf.all.forwarding=1
```

Die weitere Konfiguration ist analog zu IPv4 vorzunehmen. Da nur zwei direkt angeschlossene Netze verwendet werden, ist eine aktives Setzen der Routen nicht notwendig. Der Router kennt bereits alle Netze, da er in jedem Subnetz selbst eine Schnittstelle hat. ([s. stat. Konfigurationsbeispiele](#))

### Router Advertisement konfigurieren (stateless)

In der zweiten Übung wird zusätzlich noch das Router Advertisement benötigt, dieses kann mittels des Paket radvd installiert werden. Das Paket lässt sich mit dem folgenden Befehl installieren:

```
sudo apt-get install radvd
```

Der radvd wird standardmäßig über die Datei /etc/radvd.conf konfiguriert. Der folgende Abschnitt stellt eine rudimentäre Konfiguration dar.

[radvd.conf](#)

```
interface eth0 {  
    AdvSendAdvert on;  
    prefix 2000:db8:2::/64 {  
        AdvOnLink on;  
        AdvAutonomous on;  
        AdvPreferredLifetime 900;
```

```
    AdvValidLifetime 1500;
};
    AdvDefaultLifetime 1500;
};
```

Die Konfigurationsdatei lässt sich durch den folgenden Befehl **auf FEHLER überprüfen**:<sup>3)</sup>

```
sudo radvd -c /PFAD/radvd.conf
```

Mit -C<sup>4)</sup> kann der Daemon mit einer alternativen Konfiguration gestartet werden:

```
sudo radvd -C /PFAD/radvd.conf
```

Der Daemon kann über das folgende Kommando gestartet werden, wenn die Standard-Konfiguration (/etc/radvd.conf) verwendet wird.

```
sudo service radvd start
```

Auf der folgenden Seiten finden sich nützliche Hinweise zum radvd-Daemon:

- [andere Konfigurationsbeispiele und Debugging](#)
- [Consolen-Schalter für den Start des "radvd"](#)

## DHCPv6-Server konfigurieren (stateful)

Für die dritte Übung wird zusätzlich ein DHCPv6-Server benötigt, dieser kann mittels des Paket `isc-dhcp-server` installiert werden. Das Paket lässt sich mit dem folgenden Befehl installieren:

```
sudo apt-get install isc-dhcp-server
```

Der `isc-dhcp-server` wird standardmäßig über die Datei `/etc/dhcp/dhcp.conf` konfiguriert. Der folgende Abschnitt stellt eine rudimentäre Konfiguration dar.

### [dhcp6.conf](#)

```
# Subnetz Deklaration
subnet6 2000:db8:0:15::/64{
    range6 2000:db8:0:15::/64;
    # Einem Client eine IPv6-Adresse zuweisen
    host v6training {
        host-identifizier option dhcp6.client-id
00:01:00:01:12:ab:5d:bb:b8:b9:3e:03:32:82;
        fixed-address6 200:db8:0:15::20;
    }
}
```

Die DHCPv6-Client-ID sowie die IPv6-Adresse ist entsprechend anzupassen.

Um die Leases abzuspeichern wird eine eigene Datei benötigt. Diese wird standardmäßig unter `/var/lib/dhcp/dhcpd6.leases` abgelegt. Diese ist zunächst mit dem folgenden Befehl anzulegen.

```
touch /var/lib/dhcp/dhcpd6.leases
```

Mit `-cf` kann der Daemon mit einer alternativen Konfiguration gestartet werden. Der Parameter `-6` gibt die IP-Version an:

```
sudo dhcpd -6 -cf /PFAD/dhcp.conf
```

Der Daemon kann über das folgende Kommando gestartet werden, wenn die Standard-Konfiguration (`/etc/dhcp.conf`) verwendet wird.

```
sudo service dhcp start
```

Auf der folgenden Seiten finden sich nützliche Hinweise zum `isc-dhcp-server`:

- [andere Konfigurationsbeispiele und Debugging](#)
- [Anleitung für IPv4](#)
- [Consolen-Schalter für den Start des "isc-dhcp-server"](#)

## Der Client

Auf dem Client muss nichts Wesentliches konfiguriert werden, allerdings passieren hier die „spannenden“ Dingen.

Im Wireshark werden folgende Filter benötigt:

```
icmpv6.type==135 or icmpv6.type==136
```

Den Neighbor Cache des Clients kann man mit dem folgenden Befehl löschen:

```
ip -6 neigh flush dev eth0
```

Im Wireshark lässt sich nun eine `ping6` auf den Router analysieren. Was genau passiert nun?

Um die Router Advertisements auf dem Clients zu analysieren, kann man den folgenden Befehl verwenden.

```
rdisc6 eth0
```

**HINWEIS:** Ggf. ist das Paket `ndisc6` zu installieren. Dieses enthält das `rdisc6` <sup>5)</sup> Tool.

```
sudo apt-get install ndisc6
```

Durch De-/aktivieren des betreffenden Interfaces kann ein neues Router Advertisement angestoßen werden.

# Link-Sammlung zu IPv6-Adressen

## Allgemeine Informationen

[Glossar](#) rund um IP-Adressen. Bereitgestellt von der APNIC.

[Liste aller RFCs zu IPv6](#)

[IPv6-ready LIRs in den einzelnen Ländern](#)

[IPv6-whois Suchmaschine](#) mit der man den Besitzer/Provider einer IPv6-Adresse ermitteln kann.

## IPv6-Adresstypen

[RIPE IPv6-Adresstypen als zwei Seiten Tabelle](#)

[Beispiel für Vodafone-Adressraum](#) Das Tool kann für beliebige IPv6-Adressen angewendet werden und zeigt jeweils die Belegung des entsprechenden Adressbereichs.

[IPv6-Adresstypen von Microsoft.](#)

1)

**Router Advertisement Daemon**

2)

vgl. 2001:db8::/32 Documentation-Netz

3)

mit PFAD ist der Speicherort der Datei gemeint

4)

**ACHTUNG: großes C!**

5)

rdisc6 - ICMPv6 Router Discovery tool

From:

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

Permanent link:

[http://www.kopfload.de/doku.php?id=lager:lok\\_netze:ipv6main&rev=1386064644](http://www.kopfload.de/doku.php?id=lager:lok_netze:ipv6main&rev=1386064644)

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

