



# IPv6 Grundlagen

in Zusammenhang mit IPoAC

**Susanne Schütze**  
ratten@buecherratten.in-berlin.de

Chemnitzer Linux Tage 2024

16. März 2024

# Inhalt

▶ Kernelparameter

```
1 sysctl -w net.ipv6.conf.all.disable_ipv6=0
2 sysctl -w
  ↪ net.ipv6.conf.<Interface>.disable_ipv6=0
3 sysctl -w net.ipv6.conf.all.accept_ra=1
```

▶ systemd-networkd

```
1 [Network]
2 LinkLocalAddressing=ipv6
3 IPv6AcceptRA=yes
```

▶ Network-Manager

```
1 nmcli connection modify <ConactionName>
  ↪ ipv6.method "auto"
2 nmcli connection <ConnectionName> up
```

▶ ...

## RFC 5952

- ▶ Führende Nullen im Block weglassen

## Beispiele

```
2001:0db8:0000:0000:0000:0000:0000:0001
```

```
2001:db8:0:0:0:0:0:1
```

- ▶ längste Blöcke von Nullen durch :: ersetzen

## Beispiele

```
2001:db8:0:0:0:0:0:1
```

```
2001:db8::1
```

- ▶ mehrere Blöcke gleich lang, nur ersten von links mit :: ersetzen

## Beispiele

```
2001:db8:0:0:4ad:0:0:1
```

```
2001:db8::4ad:0:0:1
```

▶ Kernel:

```
1 sysctl -w net.ipv6.conf.all.use_tempaddr=0
2 sysctl -w net.ipv6.conf.all.addr_gen_mode=0
```

▶ Network-Manager:

```
1 nmcli con modify <ConnectionName>
  ↪ ipv6.addr-gen-mode eui64
```

▶ systemd-networkd:

```
1 [Network]
2 LinkLocalAddressing=ipv6
3 IPv6LinkLocalAddressGenerationMode=eui64
```

# Privacy-Extension

unter Linux einstellen

▶ Kernel:

```
1 sysctl -w net.ipv6.conf.all.addr_gen_mode=0
2 sysctl -w net.ipv6.conf.eth0.use_tempaddr=2
```

▶ Network-Manager:

```
1 nmcli con modify <ConnectionName>
  ↪ ipv6.ip6-privacy 2
2 nmcli connection <ConnectionName> up
```

▶ systemd-networkd:

```
1 LinkLocalAddressing=ipv6
2 IPv6PrivacyExtensions=true
```

# Semantisch undurchsichtige Interface Identifier

Unter Linux einstellen:

Im Kernel implementiert seit 2015

▶ Kernel:

```
1 sysctl -w net.ipv6.conf.all.addr_gen_mode=3
2 sysctl -w
  ↪ net.ipv6.conf.<Interface>.stable_secret=<SecretWert>
  ↪ #Secret festlegen
3 sysctl -w net.ipv6.conf.<Interface>.addr_gen_mode=2
  ↪ #Secret benutzen
```

▶ Network-Manager:

```
1 nmcli con modify <ConnectionName> ipv6.ip6-privacy 0
2 nmcli con modify <ConnectionName> ipv6.addr-gen-mode
  ↪ stable-privacy
3 nmcli connection <ConnectionName> up
```

▶ systemd-networkd:

```
1 [Network]
2 LinkLocalAddressing=ipv6
3 IPv6LinkLocalAddressGenerationMode=stable-privacy
4 IPv6StableSecretAddress=<IPv6Adresse>
```

# Multicast Adressen

## Beispiele bekannter Multicast Adressen

- ▶ ff01::1 - All Nodes Interface lokal (loopback)
- ▶ ff01::2 - All Routers Interface lokal (loopback)
- ▶ ff02::1 - All Nodes Link lokal (Bsp. Broadcast)
- ▶ ff02::2 - All Routers Link lokal
- ▶ ff02::5-6 - All OSPF Routers Link lokal
- ▶ ff02::7-8 - All IS-IS Routers Link lokal
- ▶ ff02::9 - All RIP Routers Link lokal
- ▶ ff05::2 - All Routers Sitelokal
- ▶ ff05::1:3 - All DHCP-Server in der Site

weitere Multicast Adressen:

```
https://www.iana.org/assignments/  
ipv6-multicast-addresses/  
ipv6-multicast-addresses.xhtml
```



# Das Test-Netzwerk

## IP-Adressen checken und pingen

- ▶ IP-Adresse überprüfen

```
1 ip -6 address show
```

- ▶ pingen

```
1 ping -6 fe80::e65f:1ff:fe46:edc8%<zone>
```

```
2 ping -6
```

```
↪ fd00:feed:c0ff:ee01:e65f:1ff:fe46:edc8
```

```
3 ping -6 ff02::1%<zone> # alle Nodes local
```

```
4 ping -6 ff02::2%<zone> #alle Router local
```

- ▶ **Vorsicht** man kann Unicast Adressen auch ohne Zone pingen

▶ ssh

```
1 ssh dove@fe80::e65f:1ff:fe46:edc8%<zone>
```

```
2 ssh dove@fd00:feed:c0ff:ee01:e65f:1ff:fe46:
  ↪   edc8
```

▶ hier ist die **Zone wichtig** bei LLA's

# Webseiten über IPv6

mit curl

```
1 curl -I  
  ↪ "http://[fe80::e65f:1ff:fe46:edc8%<zone>]"  
2 curl -I "http://[fd00:feed:c0ff:ee01:e65f:1ff: ]  
  ↪ fe46:edc8]"
```

Informationen aus dem Neighbour Discovery Cache abrufen:

- 1 ip -6 maddress show *#multicast Adressen*
- 2 ip -6 neighbour show *#aktueller Neighbour Cache*
- 3 ip -6 n get <ip> dev <interface> *#Informationen aus*  
↪ *Cache zu Adresse auf Interface*
- 4 ip -6 n show unused *# alle unbenutzten Nachbarn*
- 5 ip -6 n show to <IP> *#Informationen zu Adresse*
- 6 ip -6 n show dev <Interface> *#Informationen zu*  
↪ *Adressen im Interface*
- 7 ip -6 route show table all *#IPv6 Routen*

NUD einsehen unter Linux:

- 1 ip -6 neighbour show nud all *#Status der NUD im*  
↪ *Cache*
- 2 ip monitor *#Statusänderung*

### DAD Konfiguration im Kernel

- 1 `sysctl -w net.ipv6.conf.accept_dad=0 #DAD`  
→ *disablen*
- 2 `sysctl -w net.ipv6.conf.accept_dad=1 #DAD`  
→ *enablen*
- 3 `sysctl -w net.ipv6.conf.accept_dad=2 #DAD`  
→ *disablen wenn Doppelte Adresse gefunden und*  
→ *Interface Identifizier auf MAC-Adresse*  
→ *basierend*

Kontaktmöglichkeiten:

email: ratten@buecherratten.in-berlin.de

jabber: buecherratten@jabber.in-berlin.de

Repro:

<http://git.tuxteam.de/gitweb/?p=susannes-git/Ipv6-Vortrag.git;a=tree>