

>> IPv6 Leitfaden

[Bitte Kapitel auswählen]

1. Kernel einrichten

1.1 Grundlegende Kernel-Konfiguration

Jeder der erhältlichen 2.4-Kernel von Gentoo unterstützt IPv6 ohne Probleme. Zusätzlich haben *sys-kernel/gentoo-sources* und *sys-kernel/pfeifer-sources* die USAGI-Patches eingespielt. Dieses Patchset ist nicht obligatorisch, aber es wird laufend weiterentwickelt und implementiert viele Elemente von IPv6, die sonst nicht im Kernel enthalten sind. Um das Patchset zu benutzen, stellen Sie sicher, dass Sie 'usagi' in Ihrer USE-Variable haben, bevor sie eine dieser beiden Quellen emergen. Der 2.5.x Entwicklungs-Kernel hat ebenfalls ausgezeichnete IPv6-Unterstützung dank der Integration von USAGI-Code.

Befehlsauflistung 1: Einen Kernel emergen

```
# USE="usagi" emerge gentoo-sources
oder
# USE="usagi" emerge pfeifer-sources
```

Jetzt sind wir bereit, um im Kernel-Quellenverzeichnis mit der eigentlichen Konfiguration zu beginnen.

Befehlsauflistung 2: Den Linux-Kernel konfigurieren

```
# cd /usr/src/linux
# make menuconfig
```

Notiz

Dieser Befehl setzt voraus, dass der Symlink [/usr/src/linux](#) auf den Kernel-Quellcode zeigt, den Sie benutzen wollen.

Notiz

Stellen Sie sicher, dass [Prompt for development and/or incomplete code/drivers](#) in der Kernel-Konfiguration aktiviert ist.

Befehlsauflistung 3: 'make menuconfig' Einstellungen

```
Networking options --->
  <*> IP: tunneling
  ...
  <*> The IPv6 protocol (EXPERIMENTAL) --->

// Die IPv6-Optionen unter diesem Punkt können für manche
// Applikationen nützlich sein, sie werden jedoch für die
// Einrichtung nicht grundsätzlich benötigt.

Network device support --->
  <*> Universal TUN/TAP device driver support
// Diese Option wird nur benötigt, wenn Sie ptrtd für 6to4-Umwandlung
// benutzen wollen.
```

1.2 IPv6-Unterstützung testen

Nachdem Sie die empfohlenen Optionen aktiviert haben, kompilieren Sie den Kernel neu und starten das System mit dem neuen IPv6-Kernel. Wenn IPv6 funktioniert, sollte die loopback-Schnittstelle eine IPv6-Adresse anzeigen.

Befehlsauflistung 4: Die loopback-Schnittstelle überprüfen

```
# ifconfig lo
lo    Protokoll:Lokale Schleife
      inet Adresse:127.0.0.1  Maske:255.0.0.0
      inet6 Adresse: ::1/128  Gültigkeitsbereich:Maschine
      // Die Linie oberhalb zeigt, dass es funktioniert
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:4 errors:0 dropped:0 overruns:0 frame:0
      TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
```

```
Kollisionen:0 Sendewarteschlangenlänge:0
RX bytes:200 (200.0 b) TX bytes:200 (200.0 b)
```

Bevor wir nun weiterfahren, stellen Sie bitte sicher, dass Sie "ipv6" in Ihrer make.conf in die Liste von USE-Flags eingetragen haben. Zukünftige Pakete werden dann bei der Installation IPv6-Unterstützung automatisch einbinden.

2. Tunnel-Konfiguration

2.1 Grundlegende Konfiguration

Die meisten ISPs bieten immer noch keine native IPv6-Verbindungen an. Um diese Einschränkung zu umgehen, gibt es rund um die Welt sogenannte "tunnel broker", welche gratis IPv6-Tunnel zur Verfügung stellen. Dies erlaubt es Ihnen, alle IPv6-Verbindungen durch eine IPv4-Verbindung zu "tunneln".

Broker	Ort
Hurricane Electric	US/Kanada
Freenet6	US
IPnG	UK
BTEExact	Europa

Unten folgen Beispiele, um einen Tunnel mit zwei bekannten nordamerikanischen Brokern, Hurricane Electric und Freenet6, einzurichten.

2.2 Hurricane Electric

Hurricane Electric (kurz HE) bietet gratis IPv6-Tunnel an und reserviert einen /64-Block von Adressen für Sie. Zudem können Sie reversible DNS- Einträge konfigurieren. Einen Tunnel von HE zu bekommen ist einfach: Gehen Sie auf <http://www.tunnelbroker.net> und füllen Sie das einseitige Formular aus.

Notiz

Die Registration erfordert die Angabe von Informationen wie Ihre Adresse und Telefonnummer.

Warnung

Tunnel von HE brauchen 24 Stunden bis sie aktiviert sind. Dies wird gemacht, um einen Missbrauch des Angebotes zu verhindern.

Nachdem Sie einen Tunnel zugewiesen und einen /64-er Block bekommen haben, können Sie ihre Gentoo-Maschine konfigurieren. HE bietet auch Konfigurations-Beispiele basierend auf ifconfig und iproute. Die nächsten zwei Beispiele gehen davon aus, dass Sie die folgende Konfiguration haben:

```
Lokale IPv4 Adresse      68.36.91.195
HE IPv4 Adresse          64.71.128.82
Lokale IPv6 Tunnel-Adresse 2001:470:1F00:FFFF::189
IPv6 Block                2001:470:1F00:296::/64
```

Mit net-tools und dem Befehl *ifconfig* würden Sie folgendes machen:

Befehlsauflistung 5: Konfiguration mit ifconfig

```
# ifconfig sit0 up
# ifconfig sit0 inet6 tunnel ::64.71.128.82
// Einen Tunnel zur HE IPv4-Adresse hinzufügen
# ifconfig sit1 up
# ifconfig sit1 inet6 add 2001:470:1F00:FFFF::189/127
# route -A inet6 add ::/0 dev sit1
// Allen IPv6-Verkehr über die 'sit1'-Schnittstelle leiten
```

Mit dem iproute-Paket und dem *ip* Befehl würden Sie das Folgende machen:

Befehlsauflistung 6: Konfiguration mit ip

```
# ip tunnel add sixbone mode sit remote 64.71.128.82 local 68.36.91.195 ttl 255
// Einen Tunnel zwischen der lokalen IPv4 und HE's IPv4-Adresse herstellen
# ip link set sixbone up
# ip addr add 2001:470:1F00:FFFF::189/127 dev sixbone
// Den Tunnel aufschalten und eine IPv6-Adresse zuweisen
```

```
# ip route add ::/0 dev sixbone
// Alle IPv6-Adressen durch unsere 'sixbone'-Tunnelschnittstelle führen
```

2.3 Freenet6

[Freenet6](#) ist ein anderer Gratis-Tunnel-Broker. Die Registration erfordert nur einen Benutzernamen und eine gültige eMail-Adresse. Freenet6 hat sich entschieden, das Tunnel-Management an eine Server/Client-Architektur zu delegieren und hierfür *tspc* entwickelt. Das Programm ist in Portage erhalten. Um es zu installieren, machen Sie folgendes:

Befehlsauflistung 7

```
# emerge freenet6
```

Nun müssen Sie Freenet6 durch das Editieren von `/etc/freenet6/tspc.conf` einrichten. Sie sollten nur die Felder `userid` und `passwd` an die von Freenet6 zugewiesenen Einstellungen anpassen müssen. Unten ist eine komplette Beispiel-Konfiguration.

Befehlsauflistung 8: tspc.conf Beispiel

```
tsp_version=1.0.1
tsp_dir=/etc/freenet6
auth_method=any
client_v4=auto
userid=IHRE_USERID
passwd=IHR_PASSWORT
template=gentoo
server=tsp1.freenet6.net
retry_delay=0
if_tunnel=sit3
```

2.4 Ihre Verbindung testen

Nachdem nun Ihr Tunnel konfiguriert ist, können Sie die Verbindung testen. Der einfachste Weg dafür ist, das *ping6* Kommando zu benutzen und zu versuchen, einen IPv6 Rechner zu pingen.

Befehlsauflistung 9: Die Verbindung testen

```
# emerge iputils
# ping6 www.kame.net
PING www.kame.net(orange.kame.net) 56 data bytes
64 bytes from orange.kame.net: icmp_seq=1 ttl=52 time=290 ms
64 bytes from orange.kame.net: icmp_seq=2 ttl=52 time=277 ms
64 bytes from orange.kame.net: icmp_seq=3 ttl=52 time=280 ms
64 bytes from orange.kame.net: icmp_seq=4 ttl=52 time=279 ms
64 bytes from orange.kame.net: icmp_seq=5 ttl=52 time=277 ms

--- www.kame.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4038ms
rtt min/avg/max/mdev = 277.040/281.041/290.046/4.699 ms
```

Es wird momentan daran gearbeitet, bessere IPv6-Unterstützung zu den init-scripts hinzuzufügen. Wenn Sie den Status dieses Projektes wissen möchten und/oder mithelfen wollen, schreiben sie eine eMail an latexer@gentoo.org (in Englisch).

3. IPv6-Unterstützung in Applikationen

3.1 Pakete neu emergieren

Wenn Sie `USE="ipv6"` nicht schon vorher in ihrer `/etc/make.conf` hatten, müssen Sie möglicherweise einige Pakete neu kompilieren, um diese IPv6-tauglich zu machen. Um eine Liste aller installierter Pakete zu erhalten, welche die 'ipv6' USE-Variable benutzen, führen Sie das untenstehende Kommando aus.

Befehlsauflistung 10: Kandidaten für eine neu-Installation

```
# emerge -epv world | grep ipv6 | sed "s:^\.*\] ::; s:-[0-9]\+.*::" | cut -d' ' -f1
```

Um ausgewählte Pakete neu zu installieren, machen Sie folgendes:

Befehlsauflistung 11: Kandidaten für eine neu-Installation

```
# emerge -epv world | grep ipv6 | sed "s:^.*\] ::; s:-[0-9]\+.*::" | \
> cut -d' ' -f1 > packages
```

Als nächstes editieren Sie die Datei `packages` und entfernen alle Pakete, welche Sie zum jetzigen Zeitpunkt nicht neu kompilieren wollen. Danach führen Sie einfach folgendes aus:

Befehlsauflistung 12

```
# emerge `cat packages`
```

3.2 Spezielle IPv6-Pakete

Es gibt ein paar Pakete, die speziell für IPv6 konzipiert wurden. Die meisten finden Sie in `/usr/portage/net-misc`.

Paket	Beschreibung
net-misc/ipv6calc	Wandelt eine IPv6-Adresse in ein komprimiertes Format um
net-misc/nc6	Version von netcat, welche IPv6 und IPv4 unterstützt
dev-perl/Socket6	C socket.h Definitionen und Struktur-Manipulatoren für IPv6

4. DNS einrichten

4.1 IPv6 und DNS

Äquivalent zu A-Einträgen für IPv4, benutzt IPv6 AAAA-Einträge im DNS-System. (Dies weil der IPv4-Adressraum 2^{32} gross ist, während IPv6 einen Adressraum von 2^{128} hat). Für reverse DNS ist der INT-Standard der am besten unterstützte. ARPA ist das neuste Format, es ist jedoch nicht sehr verbreitet. Hier wird die Unterstützung für das INT-Format beschrieben.

4.2 BIND Konfiguration

Neuste Versionen von BIND haben ausgezeichnete IPv6-Unterstützung. Dieser Abschnitt nimmt an, dass Sie grundsätzliche Kenntnisse über Konfiguration und Nutzung von BIND haben. Wir gehen davon aus, dass Sie BIND nicht in einem chroot laufen lassen. Wenn Sie chroot benutzen, müssen Sie den folgenden Pfaden einfach Ihren chroot-Prefix voranstellen.

Als erstes müssen Sie Einträge für forward und reverse DNS-Zonen in `/etc/bind/named.conf` eintragen.

Befehlsauflistung 13: named.conf Einträge

```
zone "ipv6-rules.com" IN {
    type master;
    file "pri/ipv6-rules.com";
};
// Dies erzeugt die forward DNS für die Domäne 'ipv6-rules.com'
zone "6.9.2.0.0.0.f.1.0.7.4.0.1.0.0.2.ip6.int" {
    type master;
    file "pri/rev-ipv6-rules.com.int";
};
// Dieses Format für reverse DNS ist "bitweise". Es wird erzeugt, indem man den
// IPv6-Prefix nimmt, die Reihenfolge der Zahlen umkehrt und zwischen jede
// Zahl einen Punkt setzt.
```

Nun müssen wir die Zone-Dateien erstellen und Einträge für alle vier Hosts erzeugen.

Befehlsauflistung 14: pri/ipv6-rules.com

```
$TTL      2h
@         IN      SOA     ipv6-rules.com. webmaster.ipv6-rules.com. (
                                2003052501 ; Seriennummer (serial)
                                28800      ; Erneuerung (refresh)
                                14400      ; Wiederholung (retry)
                                3600000    ; Ablauf (expire)
                                86400 )    ; Minimum (minimum)
NS        ns1.ipv6-rules.com

IN AAAA   2001:470:1f00:296::1 ; Adresse für ipv6-rules.com
```


Wenn wir unser System als Router für andere Computer benutzen und diese mit IPv6 zur Aussenwelt Kontakt haben wollen, müssen wir die Weiterleitung von IPv6-Paketen aktivieren. Man kann dies auf zwei verschiedenen Wegen machen.

Befehlsauflistung 19: Weiterleitung aktivieren

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
oder
# sysctl -w net.ipv6.conf.all.forwarding=1
```

Warnung

Das radvd Init-Skript aktiviert Weiterleitung bereits, der nächste Schritt muss dann nicht vorgenommen werden.

Um die Weiterleitung beim Booten zu aktivieren, müssen Sie `/etc/sysctl.conf` editieren und folgende Linie hinzufügen.

Befehlsauflistung 20: Zusätzlicher sysctl.conf-Eintrag

```
net.ipv6.conf.default.forwarding=1
```

Der Verkehr sollte nun von diesem Computer durch den Tunnel an den von uns konfigurierten Broker weitergeleitet werden.

Um Clients IPv6-Adressen zuzuweisen, erlaubt die IPv6-Spezifikation sowohl statuslose (stateless) und statusbezogene (stateful) IP-Zuweisung. Statuslose Zuweisung nutzt eine Methode namens "Router Bekanntmachung" (Router Advertisement) und erlaubt Clients, eine IP und eine Route durch simples Hochfahren einer Schnittstelle zu erhalten. Es wird "statuslos" genannt, weil über die vergebenen IP-Adressen und den zugeordneten Hosts kein Protokoll geführt wird. Statusbezogene Zuweisung wird von DHCPv6 betreut. Sie heisst "statusbezogen", weil der Server den Status der Clients, die eine IP bezogen haben, speichert.

5.2 Statuslose (stateless) Konfiguration

Statuslose Konfiguration ist mit dem Router Advertisement Daemon, kurz radvd, folgendermassen zu realisieren.

Befehlsauflistung 21: radvd konfigurieren

```
# emerge radvd
```

Nachdem radvd emerged ist, müssen wir die Datei `/etc/radvd/radvd.conf` erstellen, welche Informationen darüber enthält, aus welchem IP-Block die IPs zu vergeben sind. Hier sehen Sie eine Beispiel-`radvd.conf`. Sie nutzt den Prefix, den wir vom Tunnel-Broker zugewiesen bekommen haben.

Befehlsauflistung 22: Beispiel radvd.conf

```
interface eth0
{
    AdvSendAdvert on;
// Schicke Bekanntmachungen (advertisements) an andere hosts
    AdvLinkMTU 1280;
    MaxRtrAdvInterval 300;
    prefix 2001:470:1F00:296::/64
// Die Gruppe von IPs, welche wir benutzen können.
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

Weitere Informationen sind in `man radvd.conf` enthalten. Wir können nun radvd starten und ihn zusätzlich jeweils beim Booten starten lassen.

Befehlsauflistung 23: radvd starten

```
# /etc/init.d/radvd start
# rc-update add radvd default
```

5.3 Statusbezogene (stateful) Konfiguration

Wenn Sie eine statusbezogene Konfiguration möchten, müssen Sie DHCPv6 installieren und konfigurieren.

Befehlsauflistung 24: dhcpv6 installieren

```
# emerge dhcpv6
```

Als nächstes konfigurieren wir den DHCPv6-Server indem wir `/etc/dhcp6s.conf` editieren.

Befehlsauflistung 25: Beispiel dhcp6s.conf

```
prefer-life-time 10000;
valid-life-time 20000;
renew-time 5000;
rebind-time 8000;
interface eth1 {
    link AAA {
        allow unicast;
        send unicast;
        allow rapid-commit;
        send server-preference 5;
        renew-time 1000;
        rebind-time 2400;
        prefer-life-time 2000;
        valid-life-time 3000;
        pool{
            range 2001:470:1f00:296::10 to 2001:470:1f00:296::110/64;
            prefix 2001:470:1f00:296::/64;
        };
    };
};
```

Nun können wir dhcp6s starten und so einrichten, dass er beim Booten automatisch gestartet wird.

Befehlsauflistung 26: dhcp6s starten

```
# /etc/init.d/dhcp6s start
# rc-update add dhcp6s default
```

6. IPv6 Clients

6.1 Mittels radvd

Clients hinter dem Router sollten die Möglichkeit haben, Verbindungen zum Rest des Netzes mit IPv6 herzustellen. Wenn Sie radvd benutzen, sollte die Host-Konfiguration so einfach sein wie eine Schnittstelle hochzufahren. (Dies wird möglicherweise schon durch ein net.ethX Init-Skript erledigt).

Befehlsauflistung 27: Verbinden über IPv6

```
# ifconfig eth0 up
# ifconfig eth0
eth0      Protokoll:Ethernet  Hardware Adresse 00:09:6B:06:B7:B4
inet6 Adresse: fe80::209:6bff:fe06:b7b4/64  Gültigkeitsbereich:Verbindung
inet6 Adresse: 2001:470:1f00:296:209:6bff:fe06:b7b4  Scope:Global
BROADCAST MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
Kollisionen:0 Sendewarteschlangenlänge:100
RX bytes:0 (0.0 b)  TX bytes:828 (828.0 b)
Interrupt:11 Basisadresse:0x8400 Memory:d0204000-d0204038
```

6.2 Mittels DHCPv6

Wenn Ihr Router mit DHCPv6 konfiguriert ist, müssen die Clients das dhcpv6 Paket ebenfalls installiert haben. Nachdem dies erledigt ist, müssen Sie die Clients konfigurieren indem sie die Datei `/etc/dhcp6c.conf` anpassen.

Befehlsauflistung 28: Beispiel dhcp6c.conf

```
interface eth0 {
    send rapid-commit;
    request prefix-delegation;
    request domain-name-servers;
    request temp-address;
    iaid 11111;
    renew-time 11000;
    rebind-time 21000;
};
```

7. 6to4-Umwandlung benutzen

7.1 Funktionsweise

6to4-Umwandlung kann benutzt werden, wenn Sie Hosts haben, welche mit IPv4-Computern über eine reine IPv6-Verbindung kommunizieren wollen. Mit anderen Worten: Sie haben ein reines internes IPv6-Netzwerk und einen Computer, welcher IPv4/IPv6-Verbindungen zur Aussenwelt herstellt.

7.2 DNS-Konfiguration

Um die 6to4-Umwandlung zum Laufen zu bringen, müssen Sie den DNS-Proxy *totd* konfigurieren, welcher AAAA-Einträge für Seiten erstellt, falls diese nur A-Einträge haben. Die gefälschten AAAA-Einträge zeigen dann auf IPv6-Adressen, die nicht wirklich existieren, aber dann durch den 6to4-Proxy geroutet werden. *totd* ist immer noch `~x86` maskiert, da er noch getestet wird.

Befehlsauflistung 29: *totd* installieren

```
# ACCEPT_KEYWORDS="~x86" emerge totd
```

Als nächstes müssen wir `/etc/totd.conf` mit einigen Basis-Informationen versehen.

Befehlsauflistung 30: Beispiel `/etc/totd.conf`

```
forwarder 192.168.0.2 port 53
// Zeigt auf einen richtigen DNS-Server
prefix 3ffe:abcd:1234:9876::
// Prefix für gefälschte AAAA-Einträge
port 5005
// An welchem Port totd laufen soll
pidfile /var/run/totd.pid
// Welche PID-Datei benutzt werden soll
stf
// Stichwort, um 6to4-Umwandlung einzuschalten
```

Notiz

totd muss auf einem anderen Port als 53 laufen, wenn ein anderer Namensserver auf derselben Maschine läuft.

7.3 6to4-Proxy

Wir benutzen *ptrtd* als 6to4-Proxy, welcher die Verbindungen zwischen internen IPv6-Hosts und externen IPv4-Hosts herstellt.

Befehlsauflistung 31: *ptrtd* installieren

```
# ACCEPT_KEYWORDS="~x86" emerge ptrtd
```

Wir müssen nun *ptrtd* konfigurieren und ihm sagen, für welchen Prefix er Proxy-Verbindungen erstellen soll. Editieren Sie `/etc/conf.d/ptrtd` und setzen Sie den `IPV6_PREFIX`. Dies sollte derselbe Prefix sein wie jener in der *totd*-Konfiguration.

Befehlsauflistung 32: Beispiel `/etc/ptrtd.conf`

```
IPV6_PREFIX="3ffe:abcd:1234:9876::"
```

Sie können nun *totd* starten und aktivieren, dass er beim Booten jeweils gestartet wird.

Befehlsauflistung 33: *totd* starten

```
# /etc/init.d/totd start
# rc-update add totd default
```

7.4 Clients konfigurieren und testen

Nun können die Clients konfiguriert werden, damit diese sowohl zu IPv4- als auch zu IPv6-Hosts Verbindungen über eine IPv6-Verbindung herstellen können. Wir gehen davon aus, dass die Clients schon eine IP-Adresse von radvd erhalten haben. Jetzt können wir einfach einen neuen DNS-Eintrag und die default-Route für die "falschen Adressen" erstellen. Zuerst fügen Sie zuoberst in die `/etc/resolv.conf` einen Eintrag ein, welcher auf die Maschine zeigt auf der `totd` läuft.

Befehlsauflistung 34: Beispiel `/etc/resolv.conf`

```
nameserver 2001:470:1f00:296::1
// Der Server auf dem totd läuft
```

Um die Namensauflösung zu testen, fordern Sie einen AAAA-Eintrag für eine Seite an, welche nur IPv4 kann.

Befehlsauflistung 35: Namensauflösung testen

```
# host -t aaaa google.com
google.com has AAAA address 3ffe:abcd:1234:9876::d8ef:3364
google.com has AAAA address 3ffe:abcd:1234:9876::d8ef:3564
```

Nun fügen wir eine default-Route für alle Adressen mit unserem falschen Prefix hinzu.

Befehlsauflistung 36: Benutzen des `route`-Werkzeugs

```
# route -A inet6 add 3ffe:abcd:1234:9876::/64 gw 2001:470:1f00:296::1
```

Befehlsauflistung 37: Benutzen von `iproute`

```
# ip route add 3ffe:abcd:1234:9876::/64 via 2001:470:1f00:296::1 dev eth0
```

Schliesslich können Sie nun `ping6` benutzen, um `google.com` am gefälschten IPv6-Ort anzupingen.

Befehlsauflistung 38: 6to4 testen

```
# ping6 -c 2 google.com
PING 3ffe:abcd:1234:9876::d8ef:3364(3ffe:abcd:1234:9876::d8ef:3364) 56 data bytes
64 bytes from 3ffe:abcd:1234:9876::d8ef:3364: icmp_seq=1 ttl=255 time=0.106 ms
64 bytes from 3ffe:abcd:1234:9876::d8ef:3364: icmp_seq=2 ttl=255 time=0.090 ms

--- 3ffe:abcd:1234:9876::d8ef:3364 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.090/0.098/0.106/0.008 ms
```

Notiz

Eigentlich antwortet der Router auf dem `ptrtd` installiert wurde, aber die Antwort sagt uns wenigstens, dass die Dinge richtig laufen.

8. Weitere Dokumentationen

Es gibt viele ausgezeichnete Online-Ressourcen, die sich mit IPv6 beschäftigen.

Website	Beschreibung
www.ipv6.org	Generelle IPv6-Informationen
www.linux-ipv6.org/	USAGI Projekt
www.deepspace6.net	Linux/IPv6-Seite
www.hs247.com	Alle Dinge sind IPv6
www.ipv6-net.org	Deutsche IPv6-Seite
www.kame.net	*BSD Implementation

Im IRC, können Sie `#gentoo-ipv6` oder `#ipv6` im [Freenode](http://freenode.net)-Netzwerk probieren. Sie können mit einem IPv6-fähigen Client zu den Freenode-Servern auch unter `irc.ipv6.freenode.net` verbinden.

