

IPv6-Zugang fürs LAN nachrüsten

Zweit-Router bringt IPv6 ins eigene LAN

Reiko Kaps - 16.06.11



Wer echtes IPv6 per DSL-Leitung haben will, hat aktuell nur wenig Auswahl. Besser wird es erst gegen Jahresende, wenn die Telekom bundesweit auch Privatkunden-Anschlüsse mit IPv6 versorgen wird. Bis dahin helfen nur Übergangstechniken, von denen die IPv6-Tunnel sogenannter Tunnelbroker (Gogo6, Sixxs) sehr gut funktionieren. Andere wie 6to4 oder Teredo laufen instabil und erzeugen mehr Probleme als sie lösen.

Im Folgenden zeigen wir, wie man mit einem zweiten zusätzlichen Linux-Router einen IPv6-Tunnel einrichtet, das lokale Netz mit Routing- und Präfix-Informationen für das Protokoll versorgt und das LAN mit einer einfachen IPv6-Firewall vor dem Zugriff aus dem IPv6-Internet schützt. Dieser IPv6-Router läuft in einem per NAT abgetrennten LAN, also hinter einem bereits vorhandenen Internet-Router, der die Internetverbindung per IPv4 aufbaut, Anfragen ins IPv4-Internet leitet und sich über das Domain Name System um die Namensauflösung kümmert.

Für den IPv6-Router muss man nicht gleich einen zusätzlichen Linux-Rechner aufsetzen. Für erste Experimente reicht schon eine virtuelle Maschine, die sich über eine Netzwerkbrücke ins lokale Netz einklinkt. Schöner und vor allem stromsparender erledigen billige Router wie D-Links **DIR-300[1]** (Revision A) diese Aufgabe, die man bei eBay bereits ab 10 oder 15 Euro bekommt. Hat man noch einen alten wie den **Linksys WRT54(GL)[2]** im Schrank, kann auch er als zusätzlicher IPv6-Router im LAN arbeiten. Einzige Voraussetzung: Auf dem Gerät lässt sich ein Router-Linux wie **OpenWRT[3]** installieren.

Linux als IPv6-Router

Für den Betrieb des Routers reicht ein einfaches, schlankes Linux mit einem aktuellen 2.6er Kernel, der möglichst keine Dienste wie Webserver oder Datenbanken betreiben muss. Für die Experimente setzen wir auf ein minimales, nur mit unbedingt nötigen Paketen bestücktes Debian-Linux.

Vor dem Einrichten des IPv6-Routers muss man sich bei einem Tunnelbroker einen Zugang sowie einen Netzwerk-Präfix besorgen. Sowohl Gogo6/Freenet6 als auch Sixxs bieten kostenlos solche Tunnel an und vergeben auch Präfixe, mit denen sich Rechner im eigenen LAN mit global gültigen IPv6-Adressen versorgen lassen. Beide Betreiber verlangen allerdings eine Anmeldung.

Beim Sixxs beantragt man den Zugang über ein **Webformular[4]**, das auch einige Gründe für die Nutzung abfragt. Das Netzwerkpräfix teilt das Sixxs aber erst zu, wenn der IPv6-Tunnel bereits einige Zeit läuft. Dabei honoriert das Projekt laufende IPv6-Tunnel mit Bonuspunkten (IP SixXS Kredit (ISK)), über die man etwa Support erhält oder das Präfix beantragen kann. Diese Zuteilung kann einige Tage dauern. Unter einem Debian-Linux installiert man den Sixxs-Client `aiccu` sowie den fürs IPv6-Routing zuständigen `radvd` via `aptitude`
`install aiccu radvd.`

Da der Tunnel-Client `aiccu` die Systemzeit überprüft und bei Abweichungen seinen Dienst quittiert, sollte ein NTP-Client wie `ntpd` oder `chronyd` in regelmäßigen Abständen die Uhr des IPv6-Routers über das Internet justieren. Beide Programme stehen ebenfalls im Debian-Software-Repository zur Installation mit `aptitude` bereit.

Setzt man beim IPv6-Tunnel auf Sixxs (`aiccu`), ergänzt man nach der Installation die `Aiccu`-Konfiguration unter `/etc/aiccu.conf` um den User-Namen, das Passwort, eine Tunnelkennung sowie einige weitere Optionen:

```
username SIXXS-USERNAME
password SIXXS-PASSWORD
tunnel_id TXXXX
server tic.sixxs.net
automatic true
verbose true
daemonize false
behindnat true
defaultroute true
```

Dass der IPv6-Router in einem per Network Address Translation (NAT) abgetrennten lokalen Netz arbeitet, zeigt man mit der Option `behindnat true` an. Mit `automatic true` aktiviert `aiccu` den Tunnel automatisch. Per Vorgabe tunnelt der Client die IPv6-Pakete über ein virtuelles Netzwerk-Interface (TUN-Device) `names aiccu`. Taucht das nicht in der Liste der Netzwerk-Interfaces auf (`ifconfig`), erzwingt die Option `ipv6_interface aiccu` die Namensgebung. Mit dem Konfigurationseintrag `defaultroute true` setzt `aiccu` die Standardroute für IPv6.

Anschließend startet der Tunnel-Client mit dem Kommando `aiccu start`. Mögliche Fehlermeldungen lassen sich mit dem Befehl `aiccu test` untersuchen. Steht die Option `daemonize` wie oben geschrieben auf `false`, bricht man `aiccu` mit der Tastenkombination `Strg+C` ab.

Treten beim Verbindungsaufbau keine Fehler auf, setzt man in der Konfigurationsdatei `verbose` auf `false` und `daemonize` auf `true`. Das Programm startet nun als schweigsamer Hintergrundprozess und lässt sich mit dem Befehl `aiccu stop` beenden.

Hat das Sixxs einen /48-Präfix zugewiesen, erweitert man die IPv6-Konfiguration des neuen Routers: Wie im Online-Artikel **IPv6-Internetzugang[5]** auf *heise Netze* detailliert beschrieben, erstellt man in der Datei `/etc/network/interfaces` eine neue Schnittstellenbeschreibung für das Tunnel-Interface des Sixxs-Clients. Sie kümmert sich um die Einrichtung der IPv6-Adresse für das LAN-Interface (meist `eth0`), startet und stoppt `aiccu` und setzt gegebenenfalls zusätzliche Routen.



Der IPv6-Router nutzt den vorhandenen IPv4-Internetzugang für seinen Tunnel, in dem IPv6-Pakete zum Einwahlnoten des Tunnelbrokers gelangen. Der reicht sie an die Ziele im IPv6-Internet weiter.

Läuft das Interface und der Tunnel, lässt sich die Verbindung mittels `ping6 www.heise.de` überprüfen. Antwortet der Server, funktionieren Tunnel und Namensauflösung vom IPv6-Router aus. Nun muss der Router das Präfix und die Routing-Informationen an die anderen Rechner im lokalen Netz bekannt geben (Router Advertisement, RA), was bei einem Sixxs-Tunnel einige Einträge in der Datei `/etc/radvd.conf` erledigen:

```
interface eth0 {
    AdvSendAdvert on;
    prefix 2001:db8::/64
    {
    };
};
```

Ändern Sie in diesem Beispiel den Wert hinter `prefix` auf das vom Sixxs zugewiesene Netzwerk-Präfix. Das Programm `radvd` verteilt allerdings nur /64-Präfixe, sodass man das vom Sixxs zugewiesene /48er Präfix etwa mit Nullen auffüllt. Die Zeile `interface eth0` weist `radvd` an, die Informationen über die Netzwerkkarte `eth0` raus zugeben. Mit der Option `AdvSendAdvert on` sendet der Router in Intervallen RA-Nachrichten ins Netz und antwortet auf Router-Solicitation-Anfragen von IPv6-tauglichen Clients. Außerdem muss der IPv6-Router Netzwerkpakete für IPv6 weiterreichen (Paket-Forwarding), da ansonsten `radvd` nicht startet. Das Kommando

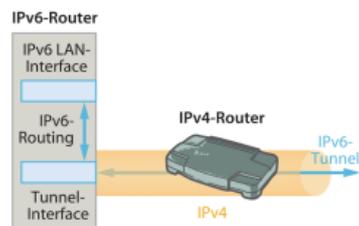
```
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

schaltet dieses Verhalten für alle Netzwerkkarten ein. Nach dem Start von `radvd` auf dem IPv6-Router sollten nun andere IPv6-taugliche LAN-Rechner wenigstens eine global gültige IPv6-Adresse mit dem eigenen Präfix erhalten, was sich unter Windows mit dem Befehl `ipconfig` oder auf Linux und Mac OS mit `ifconfig` erfahren lässt.

Go, go!

Gogo6-Tunnel funktionieren zwar auch ohne Zugangsdaten, doch bekommt man dann kein statisches Präfix zugeteilt, über das man dauerhaft seine eigenen Rechner im LAN ansprechen kann. Die Anmeldung für einen Zugang erledigt man über ein **Web-Formular[6]**, das für Passwort und Benutzererkennung nur Buchstaben und Bindestriche akzeptiert.

Nach erfolgreicher Anmeldung verschickt Gogo6 die Zugangsdaten nochmals an die angegebene E-Mail-Adresse. Das Präfix erhält man dann automatisch bei der Einwahl des Tunnel-Clients, der sich diese Informationen über das Tunnel Setup Protocol besorgt. Auf einem Debian-Linux reicht für die Installation des Gogo6-Clients das Kommando `aptitude install gogoc`, welches zusätzlich den fürs Routing zuständigen `radvd` und gegebenenfalls weitere Abhängigkeiten mit installiert. Andere Linuxe führen den Gogo6-Client unter dem Namen `gw6c`.



Der IPv6-Router transportiert über ein einziges Ethernet-Kabel sowohl IPv6- als auch IPv4-Pakete.

Tunnelt man über Gogo6 ins IPv6-Internet, kann man sich die Einrichtung von `radvd` sparen. Der Tunnel-Client `gw6c` erfährt über das Tunnel Setup Protocol (TSP) etwa das Präfix und erstellt daraus automatisch eine Konfiguration für `radvd`. Seine Konfigurationsparameter erwartet `gw6c` unter Debian in der Datei `/etc/gogoc/gogoc.conf`, die man bei einem Tunnel mit Anmeldung um Userid und Passwort erweitert. Der Eintrag `server` muss auf `authenticated.freenet6.net` zeigen.

Die automatische Konfiguration von `radvd` braucht als `host_type` den Wert `router`, die Länge des von Gogo6 zugewiesenen Präfixes (`prefixlen=56`) sowie eine LAN-Schnittstelle (`if_prefix=eth0`), an der `radvd` das Präfix verteilt. Im Unterschied zum Sixxs verteilt Gogo6 nur /56-IPv6-Präfixe, mit denen sich aber immer noch 256 eigene Subnetze bilden lassen. Debian startet nun den Client über den Befehl `/etc/init.d/gogoc start`, der den IPv6-Tunnel und den entsprechend eingerichteten `radvd` hochfährt.

Zentrale Zugangskontrolle

Wenn der Tunnel läuft und der IPv6-Router das Präfix im Netz verteilt, lassen sich alle IPv6-Computer im lokalen Netz über das Internet erreichen. Will man das nicht, kann man entweder auf jedem LAN-Rechner eine eigene IPv6-Firewall einrichten oder man blockiert den Zugang zum LAN schon im Router über den Paketfilter `ip6tables`. Läuft auf dem Router ein aktueller Linux-Kernel, reichen zwei Befehle, um alle nicht aus dem LAN aufgebauten Verbindungen zu blockieren.

```
ip6tables -A FORWARD -m state --state NEW -i eth0 -o sixxs -s 2001:db8::/64 -j ACCEPT
ip6tables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Das hinter dem Parameter `-s` stehende Quellpräfix müssen Sie unbedingt durch Ihr eigenes ersetzen. Die Anweisung leitet Pakete mit dem Quellpräfix aus dem lokalen Netz (`eth0`) über den IPv6-Tunnel (`sixxs`) ins IPv6-Internet. Mit dem Parameter `-m state --state NEW` protokolliert `ip6tables` die Pakete. Der zweite Befehl nutzt dieses Protokoll und akzeptiert nur eingehende Pakete, die sich auf die im Protokoll vermerkten beziehen. Unverlangt einlaufende verwirft der Paketfilter.

Wer etwa einen Webserver (auf Port 80) auf einer IPv6-Adresse im LAN aus dem IPv6-Netz ansprechen will, erlaubt das über eine zusätzliche Regel:

```
ip6tables -A FORWARD -i sixxs -p tcp -d 2001:db8::10 --dport 80 -j ACCEPT
```

Ersetzen Sie auch hier die Adresse hinter dem Parameter `-d` durch eine tatsächlich vorhandene aus Ihrem Netzwerk-Präfix.

Wer diese Befehle nicht jedes Mal neu eingeben will, trägt sie etwa in die Schnittstellenbeschreibung für das IPv6-Tunnel-Interface (`/etc/network/interfaces`) oder die Datei `rc.local` im Ordner `/etc` ein. Dieses einfache **Skript[7]** (ZIP-Archiv) setzt die nötigsten Firewall-Regeln und schützt so das LAN vor unerwünschten Gästen.

Zusatz-Router als IPv6-Dongle

Wie auch der DSL-Zugang sollte ein IPv6-Tunnel möglichst jederzeit laufen. Dazu eignet sich ein einfacher Linux-Router mit OpenWRT besser als ein Linux-Desktop-PC.

Die Einrichtung eines IPv6-Tunnels unterscheidet sich unter OpenWRT allerdings von der unter einem Standard-Linux. Die allermeisten Programme

konfiguriert dort entweder die Web-Oberfläche luci oder das Konsolenkommando uci, das sich um die Verwaltung von Einrichtungsdateien unterhalb von /etc/config kümmert.

Nach der **erstmaligen Installation**[8] arbeitet OpenWRT als IPv4-Router, der im LAN IPv4-Adressen per DHCP verteilt, das LAN mit einer IPv4-Firewall abschottet und sich über die WAN-Schnittstelle ins Internet verbindet. Das Root-Passwort lässt sich dabei per Telnet oder über die Web-Oberfläche setzen.

Daher hängt man den OpenWRT-Router mit seiner WAN-Schnittstelle ins bestehende lokale Netz, in dem der vorhandene Router IPv4-Adressen per DHCP verteilt. Über einen Rechner an den LAN-Ports des OpenWRT-Routers verbindet man sich nun per SSH zum Gerät, das auf die IPv4-Adresse 192.168.1.1 lauscht.

Nach dem Login deaktiviert man zuerst die IPv4-Firewall über die Befehle `/etc/init.d/firewall stop` und `/etc/init.d/firewall disable` – letzter verhindert den Start der Firewall beim nächsten Bootvorgang. Da der auf OpenWRT laufende DHCP-Server dnsmasq nur auf den zum Einrichten genutzten LAN-Schnittstellen lauscht, muss man ihn nicht zwingend abschalten.

Als nächstes aktualisiert der Befehl `opkg update` die OpenWRT-Paketdatenbank. Anschließend installiert `opkg install radvd gw6c ip6tables` alle nötigen Pakete, Abhängigkeiten wie Kernelmodule installiert `opkg` automatisch mit.

Dank seiner Unterstützung für das Tunnel Setup Protocol benötigt der Gogo6-Client gw6c nur Angaben zur Benutzerkennung und zum Einwahlserver:

```
uci set gw6c.basic.server=authenticated.freenet6.net
uci set gw6c.basic.auth_method=any
uci set gw6c.basic.userid=GOGO6-BenutzerID
uci set gw6c.basic.passwd=GOGO6-Passwort
uci set gw6c.basic.disabled=0
```

Damit gw6c den für IPv6-Routing zuständigen radvd automatisch einrichtet, muss er jedoch als Router laufen (`host_type`) und die Präfixlänge sowie die Schnittstelle fürs lokale Netz kennen. Das LAN hängt bei dieser Einrichtung abweichend an eth0.2, der sonst fürs Internet zuständigen WAN-Schnittstelle.

```
uci set gw6c.routing.host_type=router
uci set gw6c.routing.prefixlen=56
uci set gw6c.routing.ifprefix=eth0.2
```

Die mit uci gesetzten Werte speichert OpenWRT nicht sofort. Über den Befehl `uci changes gw6c` lassen sich alle Änderungen überprüfen und notfalls mit `uci revert gw6c` auf die Ausgangswerte zurücksetzen. Erst die Eingabe von `uci commit gw6c` sichert alles in die Datei `/etc/config/gw6c`.

Nach diesen Vorbereitungen startet man den Tunnel mit `/etc/init.d/gw6c start`, was gleichzeitig die Einrichtung und den Start von radvd anstößt. Der Befehl `/etc/init.d/gw6c enable` registriert den Tunnel-Client im Startsystem, sodass der Tunnel sofort beim Booten aufgebaut wird.

Für die Einrichtung eines Sixxs-Tunnels sind einige Schritte mehr nötig: Den Tunnel-Client aiccu samt radvd installiert das Kommando `opkg install aiccu radvd`. Anschließend setzen folgende Befehle die bereits genannten Einrichtungsparameter:

```
uci set aiccu.@aiccu[0].username=SIXXS-Username
uci set aiccu.@aiccu[0].password=SIXXS-Passwort
uci set aiccu.@aiccu[0].tunnel_id=TXXXX
uci set aiccu.@aiccu[0].protocol=tic
uci set aiccu.@aiccu[0].server=tic.sixxs.net
uci set aiccu.@aiccu[0].interface=sixxs
uci set aiccu.@aiccu[0].defaulttroute=1
uci set aiccu.@aiccu[0].nat=1
uci set aiccu.@aiccu[0].heartbeat=1
```

Auch den fürs IPv6-Routing zuständigen radvd füttert man über uci mit Optionen wie der LAN-Schnittstelle und dem eigenen vom Sixxs zugewiesenen Präfix:

```
uci set radvd.@interface[0].interface=wan
uci set radvd.@interface[0].AdvSendAdvert=1
radvd.@interface[0].ignore=0
uci set radvd.@prefix[0].interface=wan
uci set radvd.@prefix[0].prefix=2001:db8::/64
uci set radvd.@prefix[0].AdvOnLink=1
uci set radvd.@prefix[0].AdvAutonomous=1
uci set radvd.@prefix[0].AdvRouterAddr=0
uci set radvd.@prefix[0].ignore=0
uci set network.wan.ip6addr=2001:db8::1/64
```

Als Netzwerk-Interface für das LAN können Sie hier einfach „wan“ eintragen, die OpenWRT-Startskripte für radvd ermitteln daraus den tatsächlichen Schnittstellennamen (eth0.2). Wie auch bei den anderen Beispielen müssen Sie hinter `uci set radvd.@prefix[0].prefix=` auf jeden Fall Ihren vom Sixxs zugewiesenen Präfix eintragen. Zusätzlich braucht das fürs LAN zuständige Interface (`network.wan.ip6addr`) eine IPv6-Adresse. Die Befehle `uci commit aiccu` und `uci commit radvd` speichern die Eingaben im System. Mit den beiden Kommandos `/etc/init.d/aiccu enable` und `/etc/init.d/radvd enable` starten beide Programme beim nächsten Booten automatisch – für den sofortigen Start ersetzt man „enable“ durch „start“.

Übrigens verkauft der Tunnelbroker Gogo6 solch einen **Adapter respektive IPv6-Router**[9] für stolze 100 US-Dollar plus 20 US-Dollar Versand. Angesichts des Preises und der langen Lieferzeit dürfte jedoch ein selbst gebautes Gerät die bessere Wahl sein, um die Zeit bis zur Einführung echter nativer IPv6-DSL-Zugänge zu überbrücken. (rek)

URL dieses Artikels:

<http://www.heise.de/netze/artikel/IPv6-Zugang-fuers-LAN-nachruesten-1260260.html>

Links in diesem Artikel:

- [1] http://www.dlink.de/cs/Satellite?c=TechSupport_C&childpagename=DLinkEurope-DE%2FDLTechProduct&cid=1197319390384&p=1197318958220&packedargs=locale%3D1195806663795&pagename=DLinkEurope-DE%2FDLWrapper
- [2] http://de.wikipedia.org/wiki/Linksys_WRT54G
- [3] <https://openwrt.org/>
- [4] <http://www.sixxs.net/signup>
- [5] <http://www.heise.de/netze/artikel/IPv6-Internetzugang-221479.html>
- [6] <http://gogonet.gogo6.com/page/freenet6-registration>
- [7] <ftp://ftp.heise.de/pub/ct/listings/1108-190.zip>
- [8] <http://wiki.openwrt.org/doc/howto/basic.config>
- [9] http://gogoware.gogo6.com/4105/description.asp?product_id=180