

IPv6 Projekt & Anleitung für OpenBSD

Volker Uhrig

5. Januar 2003

Inhaltsverzeichnis

1	Einleitung	3
2	Router oder Client?	3
3	Der Tunnel	3
4	Konfigurationsdateien	4
5	Routerkonfig fürs LAN	5
6	Manuelle Clientconfiguration	6
7	Automatische Clientconfiguration	6
8	Reverse DNS	7
9	Linux	9
10	Nachwort	9

1 Einleitung

IPv6 soll irgendwann IPv4 ablösen, da der freie IPv4 Adressraum immer kleiner wird. IPv6 bietet momentan auch die Möglichkeit auch ganz nette Sachen anzustellen, vorallem bei Servern die zu Hause stehen und keine feste IP haben. Ein IPv6 Subnetz kann man als Privatnutzer über einen sogenannten TunnelBroker erhalten. Im folgenden Beispiel wird OpenBSD 3.2 auf dem Server und OpenBSD 3.1 auf dem Client verwendet. Als Tunnelbroker wird ipng.org.uk verwendet. Es empfiehlt sich beim Aufsetzen eines IPv6 Systems, die jeweiligen Manpages oder sonstige Dokumentation zu lesen. Fehler sind nützlich, sonst würde ich heute noch nicht den ganzen Spass verstehen ;)

Ein guter Start zum Verstehen von IPv6 ist die Webseite von Bieringer [5]. Im folgenden sind die Werte verändert. w, x, y, z sind eigenstehende Werte die aus den eigenen Daten zu entnehmen sind. Auch ist bei meinem Beispiel das Lokale LAN 192.168.100.0/24, was nicht überall der Fall sein muss.

2 Router oder Client?

Es besteht zunächst die Frage als was ein Rechner eingesetzt werden soll. Es besteht die Möglichkeit zwischen Router oder Host/Client.

- Router - Ein Router ist ein Rechner der Pakete weiterleitet und so als Schnittstelle für mehrere Rechner zu einem Netzwerkanschluss fungiert. Ein Router kann auch Netzwerkinformationen zur Verfügung stellen, z.B. um einem Client eine Netzwerkadresse (z.B. IPv6 Adresse) mitzuteilen.
- Client - Ein Client ist ein Rechner der in der Regel nur eine Netzwerkschnittstelle hat und als Endpunkt dasteht. Ein Client sendet seine Pakete an den Router der diese dann weiterleitet. Die Netzwerkadresse des Clients kann entweder automatisch von einem Server (oder Router) abgerufen werden oder auch statisch vorgegeben werden.

3 Der Tunnel

Ein Tunnel ist dann notwendig wenn der eigene ISP (Internet Service Provider) keine IPv6 Adressen anbietet. Hierbei wird eine Verbindung zu einem Rechner im Netz aufgebaut, durch die eine oder mehrere IPv6 Adressen oder ein komplettes IPv6 Subnet getunnelt werden.

Hierzu muss man sich erst bei einem Tunnelbroker registrieren. In diesem Beispiel wird IPNG [2] verwendet. Bei anderen Tunnelbrokern kann es zu Abweichungen hinsichtlich der Tunnelkonfiguration kommen, da andere Authentifizierungsmechanismen verwendet werden. Bei ipng.org.uk werden einem die Daten per eMail zugesendet. Es empfiehlt sich das Passwort gut zu merken, da ipng.org.uk es nicht automatisiert zustellt, was genau dann hinderlich ist wenn man es vergisst. In der eMail stehen Daten im folgenden Format (W, X, Y, Z sowie x und y an die eigenen Werte anpassen!):

```
IPv4 Endpoints: 213.253.1.201 <-> WWW.XXX.YYY.ZZZ
IPv6 endpoints: 2001:618:4:2000::xxx <-> 2001:618:4:2000::yyy
```

In die `/etc/rc.local` schreiben wir nun:

```
echo -n " ipv6"

MY_IPV6=2001:618:4:2000::yyy
TB_IPV6=2001:618:4:2000::xxx
/sbin/ifconfig gif0 inet6 $MY_IPV6 $TB_IPV6 prefixlen 128 alias
/sbin/route add -inet6 default $TB_IPV6
```

Dies setzt dann beim Neustart die IPv6 Adresse zwischen dem eigenen Router und dem Tunnelbroker auf. In `/etc/ppp/ppp.linkup` wird nun folgendes eingetragen, um ein Script bei jeder neuen ppp-Verbindung auszuführen:

```
!bg sh -c "/etc/ppp/linkupexec"
```

In der `/etc/ppp/linkupexec` steht folgendes (meinuserbeimtb und meinpassbeimtb durch die eigenen Daten ersetzen!):

```
#!/bin/sh

sleep 30
MY_IPV4='ifconfig tun0|awk '/inet / { print $2};'
TB_IPV4=213.253.1.201

# IPv6
echo -e "USER meinuserbeimtb\r\nPASS meinpassbeimtb\r\nUPIP\r\nQUIT\r\n"|nc -4 \
  ipng.org.uk 5571|grep ERR|logger -t IPv6-update

ifconfig gif0 giftunnel $MY_IPV4 $TB_IPV4
ifconfig gif0 inet6 $MY_IPV6 $TB_IPV6 prefixlen 128 alias
```

Da sich ein Nutzernamen und Passwort in diesem Skript befindet, sollten die Zugriffsrechte dieser Datei entsprechend angepasst werden (z.B. mit `chmod 700 /etc/ppp/linkupexec`). Nun sollte der Tunnel bei jeder ppp-Verbindung aufgebaut werden.

4 Konfigurationsdateien

Es gibt eine Hand voll Konfigurationsdateien die für die Router- und Clientkonfiguration notwendig sind.

- `/etc/rc.conf` - wird nur für die Routerkonfiguration benötigt
- `/etc/hostname.IF` - Statische Router und Clientkonfiguration

- /etc/hosts - übersetzt IP Adressen in Namen
- /etc/rtdadvd.conf - Konfigurationsdatei für die dynamische Adressverteilung (ähnlich dhcp)
- /etc/sysetl.conf - Routingoptionen, nur auf Seiten des Servers nötig

5 Routerkonfig fürs LAN

Wenn man sich bei ipng.org.uk einloggt (benötigt anscheinend cookies und javascript) steht unten eine Zeile mit folgendenden Daten (bitte auch nicht den Beispielerwert hier verwenden!):

```
Your IPv6 prefix is:      2001:618:4:wxyz::/64
```

Dies ist das eigene Subnetz, über deren Adressen man selbst bestimmen kann. In diesem Beispiel hat der Server das LAN Interface auf xl0, nun wird folgendes in /etc/hostname.xl0 eingetragen um eine dieser IPv6 Adressen auf das xl0 Netzwerkkinterface zu legen:

```
inet6 alias 2001:618:4:wxyz::1 64
```

Meine /etc/hostname.xl0 sieht nun folgendermassen aus:

```
taok@tree:~$ cat /etc/hostname.xl0
inet 192.168.100.1 255.255.255.0 NONE
inet6 alias 2001:618:4:wxyz::1 64
```

Nicht zu vergessen ist die /etc/syctl.conf, es ist nur von nöten

```
#net.inet6.ip6.forwarding=1      # 1=Permit forwarding (routing) of packets
```

zu entkommentieren. Dies führt dazu, das IPv6 Pakete von Clients vom Router weitergeleitet werden. Diese Konfiguration reicht vollkommen, wenn die Clients manuell konfiguriert werden. Will man jedoch den Client die Möglichkeit bieten ihre IPv6 Adresse automatisch zu erhalten, so muss der rtdadvd gestartet werden. Dies geschieht indem man in der /etc/rc.conf

```
rtdadvd_flags=NO
```

```
in
```

```
rtdadvd_flags="xl0"
```

ändert. (xl0 ist das LAN Netzwerkinterface!) Dies führt dazu das der Server beim nächsten Booten den rtadvd auf dem Netzwerkinterface xl0 startet. rtadvd braucht auch noch eine Konfiguration, die nicht weiter schwierig ist:

```
taok@tree:~$ cat /etc/rtadvd.conf
xl0:\
    :addrs#1:addr="2001:618:4:wxyz::":prefixlen#64:tc=ether:
```

Nun sollte alles funktionieren und die Clients können auch automatisch ihre IPv6 erhalten.

6 Manuelle Clientconfiguration

Der Client hat das Netzwerkinterface ne3, somit tragen wir in `/etc/hostname.ne3` folgendes ein:

```
inet6 alias 2001:618:4:wxyz::2 64
```

und in `/etc/rc.local` kommt folgendes:

```
if [ -x /sbin/route ]; then
    echo -n ' ipv6 route'; /sbin/route add -inet6 default \
    2001:618:4:wxyz::1 >> /dev/null
fi
```

Dies ist nötig, da es noch kein `/etc/mygate6` gibt und der Client wissen muss welcher Rechner als Router arbeitet und somit für die Weiterleitung der ausgehenden IPv6 Adressen zuständig ist.

Nicht vergessen sollte man das Anlegen einer `/etc/mygate`, auch wenn dies bisher noch nie benötigt wurde. Ansonsten gibt es keine Anbindung mehr für IPv6 Adressen auf dem Client. Wieso dies so ist und wo der Fehler steckt und ob er sich reproduzieren lässt ist mir bisweilen nicht bekannt, insofern empfehle ich das anlegen dieser Datei dringend!

7 Automatische Clientconfiguration

Es gibt zwei Möglichkeiten einem Client automatisch eine IPv6 Adresse zukommen zu lassen: "stateless address autoconfiguration" und DHCPv6. Beim "stateless address autoconfiguration" sendet ein Server ICMPv6 "router advertisement" Pakete, um möglichen Clients mitzuteilen das diese IPv6 Adressen erhalten können. DHCPv6 ist allerdings noch nicht in allzuvielen DHCP Servern enthalten und es wird hier auch nicht weiter darauf eingegangen. Auf dem Client wird für die "stateless address autoconfiguration" in der `/etc/sysctl.conf` folgendes entkommentiert:

```
net.inet6.ip6.accept_rtadv=1 # 1=Permit IPv6 autoconf (forwarding must be 0)
```

Dies bewirkt das der Client eine solche Autokonfiguration zulässt. Es kann schon mal vorkommen das nach dem booten des Clients keine IPv6 Adresse vorhanden ist, da vom Server noch kein ICMPv6 “router advertisement” ausgesendet wurde. Dies kann man umgehen, indem man auf dem Client `rtsol(d)`. `rtsol(d)` fragt im Netzwerk ob es eine IPv6 Adresse haben kann und wartet darauf das ein Server antwortet. Dies kann den ganzen Vorgang beschleunigen. Um den Erhalt der IPv6 Adresse per `rtsol(d)` zu beschleunigen wird folgendes in der `/etc/rc.conf` geändert:

```
rtsold_flags=NO
```

```
in
```

```
rtsold_flags="ne3"
```

(auf dem Client ist das Netzwerkinterface `ne3!`). Alternativ für den Eintrag in der `/etc/rc.conf` kann man auch in der `/etc/hostname.ne3` eine Zeile `rtsol` eintragen (siehe `hostname.if(5)`). Bei der automatischen Clientkonfiguration ist es jedoch nicht möglich wie z.B. bei DHCP die Adresse für einen Rechner anhand der MAC-Adresse vorzugeben. Dadurch erhält der Client eine schlecht merkbare Adresse. für `addr="2001:618:4:wxyz::in` in der `/etc/rtadvd.conf` z.B. `2001:618:4:wxyz:2c1:26ff:fe05:7103`.

8 Reverse DNS

Reverse DNS ist dafür da IP(v4—v6) Adressen in Hostnamen zu übersetzen. Hier soll nur die grobe Benutzung des `named` beschrieben werden. Im vorliegenden Fall wird `BIND4`, das bei `OpenBSD` dabei ist, verwendet. Abweichungen zu `BIND8` und `BIND9` liegen in der Konfigurationsdatei die beim starten des `named` gelesen wird. Um den `named` beim booten zu starten muss man in der `/etc/rc.conf` die Zeile

```
named_flags=NO
```

```
in
```

```
named_flags=""
```

ändern.

Bei `OpenBSD` sind die Konfigurationsdateien in `/var/named`. Die Hauptkonfigurationsdatei, die beim Starten des Servers gelesen wird (sofern keine andere manuell angegeben wird), ist die `/var/named/named.boot`. Diese Datei braucht zwei Zonefileeinträge, damit der Rechner weiss das er für die zwei RDNS-Zonen zuständig ist:

```
primary z.y.x.w.4.0.0.0.8.1.6.0.1.0.0.2.ip6.int 2001.618.4.wxyz.int.v6.rev
primary z.y.x.w.4.0.0.0.8.1.6.0.1.0.0.2.ip6.arpa 2001.618.4.wxyz.arpa.v6.rev
```


Um zu überprüfen ob der reverse Eintrag wirklich dem richtigen Server zugeteilt worden ist kann man die Nameserver befragen:

```
$ host -t NS 4.0.0.0.8.1.6.0.1.0.0.2.ip6.int
4.0.0.0.8.1.6.0.1.0.0.2.ip6.int NS      dangermouse.pod4.org
4.0.0.0.8.1.6.0.1.0.0.2.ip6.int NS      penfold.noc.clara.net
```

Die zuständigen Server sind also dangermouse.pod4.org und penfold.noc.clara.net. Nun fragen wir diese welcher Server für das Subnetz 2001:618:4:wxyz zuständig ist:

```
$ host -t NS z.y.x.w.4.0.0.0.8.1.6.0.1.0.0.2.ip6.int
z.y.x.w.4.0.0.0.8.1.6.0.1.0.0.2.ip6.int NS      ns.domain.tld
$ host -t NS z.y.x.w.4.0.0.0.8.1.6.0.1.0.0.2.ip6.arpa
z.y.x.w.4.0.0.0.8.1.6.0.1.0.0.2.ip6.arpa      NS      ns.domain.tld
```

Bei ipng.org.uk gibt man im Webinterface ein, welcher Nameserver rDNS für die eigenen Adressen zur Verfügung stellen soll. Allerdings hatte ich bei ipng.org.uk das Problem das zwar ein reverse ip6.arpa Eintrag vorhanden war, aber ip6.int gefehlt hat. Der Support von diesem TB erklärte mir dieses nichtvorhanden sein mit einem 'out of sync', was aber behoben wurde (die Beantwortung der eMail hat 12 Tage gedauert).

9 Linux

Eine kleine Bemerkung zu Linux IPv6 Clients. Ist IPv6 im Linuxkernel (getestet mit 2.4.20), so erhält der Client sofort automatisch eine IPv6 Adresse per rtdvd. Es gibt wohl auch eine Möglichkeit diese Autokonfiguration auszuschalten. Näheres dazu kann man in der Dokumentation der Linuxquellen finden (i.d.R. in /usr/src/linux/Documentation/networking/ip-sysctl.txt). Die Defaulteinstellung könnte allerdings auch Distributionsabhängig sein. Da auch die Konfiguration einer fest vorgegebenen IPv6 Adresse oftmals Distributionsabhängig ist, wird hier nicht weiter darauf eingegangen werden. Desweiteren verbleibt nur noch die Empfehlung auf das Linux IPv6 HOWTO von Peter Bieringer [5].

10 Nachwort

Es ist natürlich auch möglich den rDNS Server auf einem Rechner der nur über eine dynamische IPv4 Adresse hat laufen zu lassen. Hierzu kann man einen DynDNS Service verwenden (z.B. von DynDNS.org [3] oder evilroot.org [4]). Eine eigene Domain von z.B. EU.org [?] (ist kostenlos...) kann man auch auf den eigenen Rechner legen, indem man diesem zwei DynDNS Adressen zuordnet. Von einem einzigen Mailserver ist jedoch abzuraten, da DNS Adressen auch gecached werden und es auch generell zu Netzwerkproblemen kommen kann. Wenn es schon sein muss, dann ist ein Backup-MX mehr als nur zu empfehlen. Auch sollte der Primary oder Secondary NS extern ausgelagert werden. Andere Server wie z.B. Webserver sind natürlich auch möglich. Es gilt allerdings bei IPv6 Servern zu bedenken, das die wenigsten Nutzer

IPv6 fähige Rechnersysteme haben! für IRC Nutzer kann IPv6 auf einem Rechner mit ständig wechselnder IPv4 auch einen grossen Vorteil bringen. Mit vorhandenem RDNS ist es möglich ständig mit festem Hostnamen im IRC zu bleiben. Es wird zwar die IPv4 Verbindung neu aufgebaut, aber mit dieser auch der IPv6 Tunnel. Und oft geht dies so schnell, das es zu keinem Ping Timeout kommt. Aber das nur am Rande bemerkt.

Ziel einer DynDNS, NS, IPv6 und MX Spielerei ist eigentlich sich mit den einzelnen Protokollen, verschiedenen Servern und Systemen auseinanderzusetzen. Und man kann hierbei viel über das Internet lernen, wie das DNS aufgebaut ist, wie eMail überhaupt funktioniert etc. ppl..

TODO: aufteilen, einmal lang und einmal in Stichworte

Id : ipv6.tex, v1.22003/01/0517 : 18 : 49wbxExp

Internet-Quellen

- [1] *OpenBSD Projektseite*
<http://www.openbsd.org/>
- [2] *IPNG - Tunnelbroker*
<http://ipng.org.uk>
- [3] *Dyndns.org*
<http://www.dyndns.org/>
- [4] *evilroot.org*
<http://www.evilroot.org/>
- [5] *Bieringer IPv6*
<http://www.bieringer.de>