Inhaltsverzeichnis

Vorbemerkungen

Im Zuge der Umstellungen auf IPv6 habe ich ausgiebig im Web recherchiert. Dabei habe ich einige gute Informationen gefunden, manche Sachen waren aber wenig dokumentiert. Man merkt, dass das Thema noch neu ist, obwohl es den Standard bereits seit 1998 gibt.

Ich möchte hier nur Ausführungen zu Themen machen, die andere noch nicht behandelt haben. Ansonsten verweise per Link auf die Informationen. Ich hoffe damit anderen bei Problemen zu helfen, an denen ich mir (fast) die Zähne ausgebissen habe.

Was ist der Grund für den plötzlichen Hype?

Die verfügbaren Adressen im IPv4 Bereich gehen langsam zu Neige, irgendwann werden neue Hosts auf IPv6 ausweichen müssen. APNIC (Asia-Pacific Network Information Centre), der für den asiatischen und pazifischen Raum zuständig ist, meldete am 15. April 2011 "Ende Gelände". IANA (Internet Assignment Authority), zuständig für die Vergabe an APNIC und die anderen lokalen Zentren, bereits am 3. Februar 2011:

"APNIC IPv4 Address Pool Reaches Final /8" (Englisch)
www.apnic.net/publications/news/2011/final-8
"Available Pool of Unallocated IPv4 Internet Addresses Now Completely Emptied" (Englisch)
http://www.icann.org/en/news/releases/release-03feb11-en.pdf

Das ist zwar nicht gleich Grund für Panik, aber wenn man auf längere Sicht Zugriff auf alle Internet Ressourcen behalten will, sollte man sich langsam vorbereiten. Im Augenblick gibt es nur sehr wenige reine IPv6 Angebote, auf die man bequem auch verzichten kann. Auf Dauer wird es aber sicherlich interessant werden, da irgendwann neue Hosts Auftauchen werden, die keine IPv4 Adresse mehr erhalten.

Diskussion zu Stateful IPv6

Zum Thema "Stateful IPv6", also die Vergabe von IP Adressen per DHCP, finden sich diverse Diskussionen im Internet bis hin zum Kneipenniveau nach diversen Bier und Korn. Da flogen die virtuellen Fäuste und die Beleidigungen.

Für manche scheint das schon fast eine Glaubensfrage zu sein, ich fühlte mich ein bisschen an Russels Teekanne erinnert (http://de.wikipedia.org/wiki/Russells_Teekanne):

"...Leute, die ihre Milch zuerst einschenken, schießen nicht jenen, die den Tee zuerst einschenken, die Kniescheiben weg...."
übersetzt aus Richard Dawkins: A Devil's Chaplain: Selected Essays

Ganz nüchtern betrachtet, was wollen wir?

User wollen den Rechner anmachen und im Web surfen, Mails lesen, Laufwerke lokal und im Netzwerk nutzen.

Admin muss ein DNS bereitstellen, um Netzwerkressourcen auffindbar zu machen, IPs vergeben um gruppieren zu können und Zugriffsrechte zu kontrollieren, Internet Gateway

und Proxy festlegen können, um eventuell Webseiten zu sperren oder nach Viren zu scannen, und natürlich Unbefugte aussperren.

Beides lässt sich recht einfach mittels der Kombination DNS/DHCP6 bewerkstelligen. Man kann Rechnern feste IPs geben und auch ändern, ohne jedesmal durch die Flure zu latschen. Rechner können sich gegenseitig per DNS finden (bei den sehr langen IPv6 Adressen, die echt schwer zu merken sind, noch wichtiger wie zuvor). Ausserdem kann man unbefugte Rechner aussperren.

Zusätzlich lassen sich wichtige Informationen wie Gateways, Router, DNS Suffixe usw. mit übertragen.

Also ganz pragmatisch:

- Der Vorteil von Stateless Address Autoconfiguration ist, dass sie dezentral erfolgt ohne dedizierten Server.
- Der Vorteil von Stateful Address Autoconfiguration ist, dass sie zentral erfolgt mit dediziertem Server.

Deshalb sage ich: Warum nicht mit DHCP6? Das Verfahren ist erprobt, es ist gut zu beherrschen, und es ermöglicht eine gute Kontrolle.

Welche Endgeräte unterstützen überhaupt IPv6?

Die meisten Geräte unterstützen IPv6 nicht, d.h., man kann nur eine IPv4 Adresse konfigurieren, sie sind also über IPv6 nicht erreichbar. Switches und WLAN Access Points leiten die Pakete trotzdem problemlos weiter.

Bei Routern ist das aber leider nicht möglich, wenn keine direkte Unterstützung geboten wird. Geräte, die beide Protokolle mittels "Dual Stack" unterstützen sind FritzBox oder der Cisco RVS 4000.

Wie kommt man an IPv6?

Was den Zugriff auf lokale Ressourcen betrifft, namentlich die Umstellung von Windows und Unix Rechnern auf IPv6, wird das weiter unten genauer beschrieben. Interessant ist aus besagten Gründen in erster Linie der Zugriff auf das Internet über IPv6.

Direkter Zugriff: Unterstützung hat bisher nur die Deutsche Telekom zugesagt. Andere Anbieter halten sich bedeckt: www.zdnet.de/news/41538861/deutsche-telekom-bietet-ipv6-fuer-privatkunden-ab-ende-2011.htm.

Mit Hilfe eines Tunnels: Dazu gibt es einen hervorragender Artikel im deutschen ZDNET: "IPv6 mit festen Adressen: So nutzt man SixXS-Tunnel" www.zdnet.de/magazin/41522303/ipv6-mit-festen-adressen-so-nutzt-man-sixxs-tunnel.htm. Dort wird auch beschrieben, wie man einen IPv6-fähigem Anbieter nutzen kann.

Auf den Seiten von SixXS stehen weitere, wichtige Details zu statischen Ipv4-zu-IPv6

Tunneln (Englisch): www.sixxs.net/faq/connectivity/?faq=ossetup

Speziell zu SuSE 10.1 und 10.2 sollte man hier nachlesen (Englisch):
www.sixxs.net/wiki/SuSE10.1_configuration

Für SuSE 10.2 muss entgegen der dort gemachten Aussagen das Interface sit0 verwendet werden, da der Tunnel sonst nicht funktioniert.

Sicherheitsaspekte

Ein wichtiger Punkt fehlt bei der Beschreibung, wie man mittels Unix-Rechner ins Internet kommt: Die Firewall! Diese ist unabdingbar, da das IPv6 Interface nicht durch NAT und die normale Firewall geschützt ist. Weil bei Verwendung eines IPv6 Subnets alle Adressen im Web bekannt sind und geroutet werden, sind sonst alle angeschlossenen Rechner ungeschützt erreichbar!

Im Augenblick mag die Hackeraktivität noch minimal sein im IPv6 Bereich, aber das Risiko wäre einfach zu gross. Deshalb bitte unbedingt nach erfolgreichem Aufbau des Tunnels wie unten beschrieben die Firewall aktivieren.

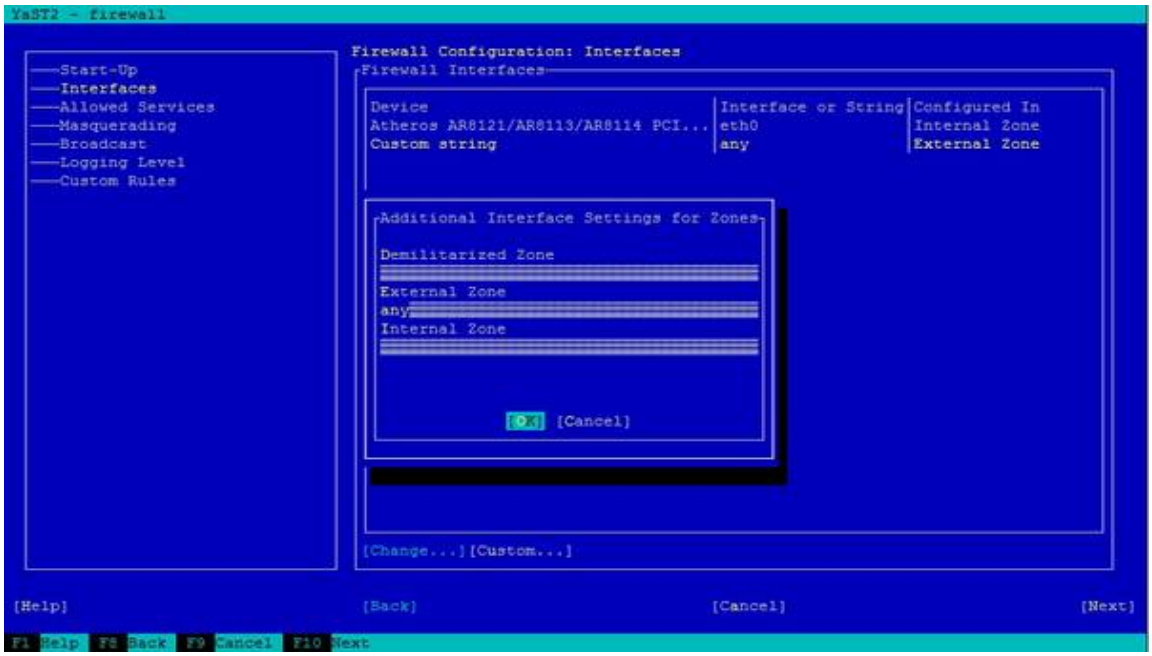
Firewall

Als erstes sollte man die Firewall aktivieren. Bei SuSE 11.x trägt man das Interface (z.B. sixx0) die Liste der externen Interfaces ein, oder sichert einfach mit "any" alle unbekannten Netzwerkadapter ab.
In der Datei /etc/sysconfig/SuSEfirewall2 muss dazu folgendes eingetragen bzw. geändert werden:

```
FW_DEV_EXT="any"
```

Die Firewall danach neu starten mit "SuSEfirewall2 stop" und "SuSEfirewall2 start".

Diese Einstellung kann man auch mit Yast ändern:



Damit das Routing funktioniert, muss ausserdem manuell das Forwarding vom internen Netz freigeschaltet werden. Sonst ist der Zugriff nur vom Rechner mit dem sixxs0 Interface möglich.

In der Datei /etc/sysconfig/SuSEfirewall2 muss dazu folgendes eingetragen bzw. geändert werden (hier mit dem Netzsegment wie im folgenden Beispiel – der erste Teil muss mit dem internen Subnetz übereinstimmen). Sind schon andere Einträge dort, ist ein Leerzeichen einzufügen und die Adressen zu ergänzen:

```
FW_FORWARD="2001:0908:0706::/48,2000::/3"
```

Die Firewall danach neu starten mit "SuSEfirewall2 stop" und "SuSEfirewall2 start".

Sollen Dienste von aussen erreichbar sein, z.B. ein Webserver, können diese beim SuSE 11.x wie IPv4 Ports über Yast freigeschaltet werden. Bei SuSE 10.x ist das nicht so einfach, da die ip6tables keine "stateful" Firewall unterstützen. Die Ports müssen einzeln manuell freigegeben werden.

Beispiel:

Das folgende Script kann mit "ip6tables < script" eingespielt werden. Es erlaubt Ping, Traceroute und gibt Port 25 und 80 für SMTP und einen Webserver frei:

```
# Generated by ip6tables-save v1.3.6 on Sun Oct 16 17:29:00 2011
# NS: Added server access rules (16.10.2011)
*mangle
:PREROUTING ACCEPT [3:212]
:INPUT ACCEPT [3:212]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [3:356]
:POSTROUTING ACCEPT [3:356]
COMMIT
# Completed on Sun Oct 16 17:29:00 2011
# Generated by ip6tables-save v1.3.6 on Sun Oct 16 17:29:00 2011
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:reject_func - [0:0]
-A INPUT -s ::/0 -d ::/0 -i lo -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 133 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 134 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 135 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 136 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 137 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 129 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 1 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 2 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 3 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 4 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p ipv6-icmp -m icmp6 --icmpv6-type 128 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p tcp -m tcp --tcp-flags ACK ACK -j ACCEPT

# Server access...
-A INPUT -s ::/0 -d ::/0 -p tcp -m tcp --dport 25 -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p tcp -m tcp --dport 80 -j ACCEPT
# End server access...

-A INPUT -s ::/0 -d ::/0 -j reject_func
```

```
-A OUTPUT -s ::/0 -d ::/0 -o lo -j ACCEPT
-A OUTPUT -s ::/0 -d ::/0 -p ipv6-icmp -j ACCEPT
-A OUTPUT -s ::/0 -d ::/0 -p tcp -j ACCEPT
-A OUTPUT -s ::/0 -d ::/0 -j reject_func
-A reject_func -s ::/0 -d ::/0 -p tcp -j REJECT --reject-with tcp-reset
-A reject_func -s ::/0 -d ::/0 -p udp -j REJECT --reject-with icmp6-port-
unreachable
-A reject_func -s ::/0 -d ::/0 -j REJECT --reject-with icmp6-addr-unreachable
-A reject_func -s ::/0 -d ::/0 -j DROP
COMMIT
# Completed on Sun Oct 16 17:29:00 2011
```

Weiterführende Details zum Finetuning sowie zur Vorgehensweise bei FreeBSD finden sich auf den folgenden Seiten (Englisch):
www.sixxs.net/wiki/IPv6_Firewalling#Example_script_for_IPv6_stateful_firewall

IPv6 Privacy Extensions

In manchen Betriebssystemen wird bei der automatischen Vergabe von IPv6 (Stateless Address Autoconfiguration, SLAAC) die MAC Adresse der Netzwerkkarte verwendet. Damit wird die Maschine im Internet identifizierbar. Zu diesem Zweck gibt es die Privacy Extensions, die mittels Zufallswerten diese Information verbergen:

www.heise.de/netze/artikel/IPv6-Privacy-Extensions-einschalten-1204783.html

Dieser Hinweis nur zur Vollständigkeit, da wir hier kein SLAAC verwenden, betrifft das Problem diese Beschreibung nicht.

Konfiguration DHCP6 und DNS

In den folgenden Kapiteln wird beschrieben, wie man DHCP6 einrichtet, damit dynamisch und fest IPv6 Adressen vergeben werden. Ausserdem, wir man eine Anbindung an den DNS Server realisiert, damit Rechner über ihren Namen bekannt sind.

Eingesetzte Zusatzsoftware

SuSE 11.4 (Server):
DHCP6 Server V4.2.1
radvd V1.3
Beides kann mittels SuSE Softwaremanagement installiert werden.

Windows XP:
dibbler V0.8.0 http://klub.com.pl/dhcpv6/ (nur als DHCP6 Client)

FreeBSD:
KAME DHCP6 Client, Server und Relay
Befindet sich als dhcp6-20080615_1 aus den FreeBSD Ports.

Windows Vista, Windows 7 und SuSE 11.4 (Client):
Es wird keine Zusatzsoftware benötigt.

Betriebssysteme

Windows XP SP3/Vista/7
SuSE 11.4 (Server)
SuSE 11.4 (Client)
FreeBSD 8.x (Client)
FreeBSD 9.x (Client)

IP-Adressen

Die folgenden Adressen werden in den Beispielen verwendet.

Die Adresse des IPv6 Routers ist:
2001:0102:0304:0506::2

Das zugeteilte IPv6 Subnet ist:
2001:0908:0706::/48

Das IPv4 Netz ist:
192.168.1/24

DNS und DHCP Server (sowie optional SAMBA):
192.168.1.1
2001:0908:0706::1

Für DHCP Clients reservierter Adressbereich:
2001:0908:0706:FFFF::

Die verwendete Domain:
intranet.mydomain.tld

Nur mal so am Rande: wir reden hier von 1.2089258196146292e+24 Hosts, die sich in meinem Subnetz tummeln könnten. Das ist über 1.2 Quadrillionen, eine Mordszahl, um das mal so zu sagen.

Da die Routeradresse nicht Teil des Subnetz ist, sind zusätzliche Massnahmen nötig, um ein Routing hinzubekommen - siehe Firewall.

Server (SuSE 11.4)

Als Server dient die SuSE 11.4 Maschine. Vermutlich wird die beschriebene Vorgensweise auch mit anderen 11.x SuSE funktionieren. Mit älteren SuSE Versionen gibt es verschiedentlich Probleme, auf die so weit bekannt gesondert hingewiesen werden.

radvd.conf

Da die Weitergabe von Router-Informationen aus dem DHCP6 Protokoll herausgeflogen ist, muss ein zusätzlicher Dienst aktiviert werden: Der Router Advertisment Daemon (radvd). Er informiert die im IPv6 Netz angeschlossen Maschinen über vorhandene Routing-Gateways. Ausserdem wird er für die statuslose Autokonfiguration benötigt, die wir aber nicht verwenden.

Damit DHCP funktioniert, müssen folgende Werte gesetzt sein:

```
AdvManagedFlag on;
AdvOtherConfigFlag on;
```

Erlaubt DHCP6 Clients, ihre Adresse von DHCP6 und weitere Informationen (DNS Server, Domain Suchpfad etc.) zu beziehen.

Für das Präfix muss ausserdem dies deaktiviert werden:

```
AdvAutonomous off;
```

Dies deaktiviert die Autokonfiguration der Clients.

In meinem Netz hat sich ausserdem als sinnvoll erwiesen, diese Werte zu setzen:

```
MinRtrAdvInterval 3;
MaxRtrAdvInterval 10;
```

Es beschleunigt die Vergabe von Adressen. Ich hatte einen Killer-Rechner, der, wenn er angeschlossen wurde, sämtliche anderen Rechner zu einem aktiven DHCP Release veranlasste. Wieso konnte ich bisher nicht genau ermitteln, nur, dass tatsächlich alle anderen ihre IPv6 Adresse beim DHCP als frei meldeten.

Mit den Standardintervallen hatten sie danach manchmal minutenlang keine IPv6 Adresse. Mit den o.g. Werte waren sie nach wenigen Sekunden wieder im Netz.

Die vollständige radvd.conf: http://www.oblivion-software.de/uploads/media/radvd.conf

Dynamic DNS

Grundsätzlich funktioniert die DNS Anbindung von DHCP6 nicht anders wie bei DHCP4. Zunächst benötigt man einen gemeinsamen Schlüssel, falls nicht schon einer für den DHCP4 existiert (den man dann mitverwenden kann), muss einer generiert werden:

```
linux-dhcp:~ #genDDNSkey --key-file /etc/named.d/dhcp_dyn_dns.key --key-name
DHCP_UPDATER
```

Sollte das Script nicht vorhanden sein, muss das **bind-utils** Paket installiert werden.

Es dauert ein bisschen – Geduld. Dies erzeugt drei Dateien:

Kdhcp_updater..key*
Kdhcp_updater..private*
dhcp_dyn_dns.key

Von denen ist **dhcp_dyn_dns.key** für uns die einzige interessante. Die K*.private kann zusammen mit nsupdate zur Aktualisierung des DNS verwendet werden. Die Beschreibung würde den Rahmen hier sprengen, mehr dazu kann mit "man nsupdate" abgerufen werden (bei SuSE 10 ist das allerdings notwendig).

Der Inhalt der Datei sieht z.B. folgendermassen aus:

```
# generated by genDDNSkey on Sat Oct 22 19:44:45 CEST 2011

key DHCP_UPDATER {
algorithm hmac-md5;
        secret
"yw6vHqOVpst+xUu//K4nQAEkR5ia75tZSj+0MYaxF6vSE0umKDpKrFo8nyEj6cc48mdP+EaVa5fevBG
l8E2RZg==";
};
```

In manchen Beispielen im Netz wird empfohlen, obigen Code einfach in dhcp.conf und named.conf zu kopieren. Ich halte es aber für effektiver, die Datei mit include einzufügen.

dhcpd6.conf

Die wichtigsten Einstellungen zusammengefasst:

Optional: Setzen eines Time-Servers (z.B. der Physikalisch Technischen Bundesanstalt), einen Druckserver, den SAMBA Domain Controller.

```
option ntp-servers ntp1.ptb.de;
option lpr-servers druckknecht1.intranet.mydomain.tld;
option netbios-name-servers samba.intranet.mydomain.tld;
```

Die lokale Domain inklusive DNS Server:

```
option domain-name "intranet.mydomain.tld";
option dhcp6.name-servers 2001:0908:0706::1;
option dhcp6.domain-search "intranet.mydomain.tld";
```

Den MD5 Schlüssel für das dynamische DNS und die Einstellungen dafür:

```
include "/etc/named.d/dhcp_dyn_dns.key";

ddns-domainname "ipv6.intranet.mydomain.tld";
ddns-update-style interim;
ignore client-updates;
ddns-updates on;
ddns-rev-domainname "ip6.arpa.";
```

Das Subnetz mit dem Adresspool:

```
subnet6 2001:0908:0706::/48 {

      # Range for clients
      range6 2001:0908:0706:FFFF::/64;
```

Nochmals dynamisches DNS:

```
      zone mydomain.tld.
      {
            primary localhost;
            key DHCP_UPDATER;
```

```
      }
      zone 6.0.7.0.8.0.9.0.1.0.0.2.ip6.arpa.
      {
            primary localhost;
            key DHCP_UPDATER;
      }
```

Festlegen von festen IPv6 Adressen für einzelne Hosts:

```
      host other-pc {
            host-identifier option dhcp6.client-id
00:01:00:01:25:89:AA:2F:00:1F:DD:A0:9C:1A;
            fixed-address6 2001:0908:0706::c0a8:010c;
      }
      host some-pc {
            host-identifier option dhcp6.client-id
00:01:00:01:0F:3A:AA:28:9D:88:72:4F:EE:A0;;
            fixed-address6 2001:0908:0706::c0a8:0111;
      }
```

Die letzten 32 Bit habe ich so festgelegt, dass sie den IPv4 Adressen des Rechners entsprechen. Das ist nicht zwingend so, aber erleichtert die Zuordnung, wenn nur die IPv6 Adresse bekannt ist:

other-pc	2001:0908:0706::c0a8:10c	192.168.1.12
some-pc	2001:0908:0706::c0a8:111	192.168.1.17

Die Vergabe funktioniert prinzipiell wie bei DHCP4, nur wird statt der Hardware Adresse (MAC Adresse) des Interface eine sogenannte "Client DIUD" verwendet. Diese wird normalerweise bei der Installation des DHCP6 Clients generiert. Siehe Hinweise bei den einzelnen Betriebssystemen weiter unten, wie man die DIUD ermittelt. Sollte der Abruf nicht so einfach möglich sein, helfen die log(debug...)-Einträge.

DHCP6 muss danach mittels Yast aktiviert werden. Im Runlevel-Editor für Level 3 und 5 auf aktiv setzen und starten.

Die vollständige dhcp6.conf: http://www.oblivion-software.de/uploads/media/dhcp6.conf

Bemerkungen

Eigenlich sollte es auch möglich sein, den DNS Server mit den statischen Leases zu aktualisieren. Dann wird die IP des Hosts ins DNS eingetragen, genau wie bei den dynamischen Leases. Davon wird zwar im RFC audrücklich abgeraten, kann jedoch sehr praktisch sind

"Es sollte gehen", wie man bei Computern oft gerne sagt. Leider klappte das bei mir gar nicht – deshalb hier die Info, wie es gehen sollte und viel Spass beim Ausprobieren:

```
update-static-leases on;
use-host-decl-names on;

subnet6 ...
      host other-pc {
            host-identifier option dhcp6.client-id
00:01:00:01:25:89:AA:2F:00:1F:DD:A0:9C:1A;
```

```
            fixed-address6 other-pc.ipv6.intranet.mydomain.tld;
      }
```

Oder mit expliziter Angabe des Hostnamen:

```
update-static-leases on;
use-host-decl-names off;

subnet6 ...
      host other-pc {
            host-identifier option dhcp6.client-id
00:01:00:01:25:89:AA:2F:00:1F:DD:A0:9C:1A;
            fixed-address6 other-pc.ipv6.intranet.mydomain.tld;
            option host-name "other-pc";
      }
```

named.conf

Zunächst müssen wir den Nameserver anweisen, auf dem neuen IPv6 Interface auf Anfragen zu hören. Natürlich sollte man den neuen lokalen Host nicht vergessen:

```
options {

      # The listen-on-v6 record enables or disables listening on IPv6
      # interfaces.  Allowed values are 'any' and 'none' or a list of
      # addresses.

      listen-on-v6 { ::1; 2001:0908:0706::1; };
};
```

Dann die lokale Zone sowie die beiden Reverse-Zonen:

```
acl intern { 192.168.0/16; };

zone "mydomain.tld" in {
      file "dyn/mydomain.tld";
      type master;
      allow-transfer { intern; localhost; localnets; };
      allow-update { key dyn_dns; };
};
zone "168.192.in-addr.arpa" in {
      allow-transfer { intern; localhost; localnets; };
      file "dyn/168.192.in-addr.arpa";
      type master;
      allow-update { key dyn_dns; };
};
zone "6.0.7.0.8.0.9.0.1.0.0.2.ip6.arpa" in {
      allow-transfer { intern; localhost; localnets; };
      file "dyn/6.0.7.0.8.0.9.0.1.0.0.2.ip6.arpa";
      type master;
      allow-update { key dyn_dns; };
};
```

Die folgende Zeile gehört bei SuSE 11.x in die Datei "**/etc/named.conf.include**", die wiederum in der named.conf eingebunden wird:

```
include "/etc/named.d/dhcp_dyn_dns.key";
```

Sie macht den eben generierten Schlüssel auch dem DNS Server bekannt.

Der Nameserver sollte danach mit "/etc/init.d/named restart" neu gestartet werden.

DNS Einträge für die dynamisch verwalteten Zonen (mit Journal-Dateien – jnl) müssen danach mittels Yast angepasst werden, keinesfalls mehr direkt in den Zonen-Dateien.

Wichtiger Hinweis für Suse 10.x:
Die mitgelieferte Yast Version unterstützt die Handhabung von dynamischem DNS nicht. Es darf danach nicht mehr verwendet werden, um DNS zu konfigurieren (es entfernt jedesmal die Einbindung der Schlüssel).

DNS Einträge müssen dann manuell in den Zonen-Dateien bzw. für die dynamisch verwalteten (mit Journal-Dateien – jnl) mittels "nsupdate" angepasst werden. Die Beschreibung würde den Rahmen hier sprengen, mehr dazu kann mit "man nsupdate" abgerufen werden.

Die vollständige named.conf: http://www.oblivion-software.de/uploads/media/named.conf

Zusammenfassung

Der Server ist hiermit konfiguriert, DHCP6 und DNS laufen und sind verbunden. Der Einrichtung der Clients steht damit nichts mehr im Wege.

Clients

Windows 7/Windows Vista

Diese Windows-Versionen unterstützen IPv6 bereits vollständig "aus der Schachtel". Es muss nichts weiter eingerichtet werden.

Tip:

Ein erzwungener Refresh der IP lässt sich per Kommandozeile auslösen:

ipconfig /renew6

Bestimmung der DIUD

Die Client DIUD lässt sich bei Windows ganz bequem über die Kommandozeile abfragen:

ipconfig /all

Die ID mit ihren 14 Bytes steht dann in der Zeile "DHCPv6-Client-DUID".

Windows XP

Bei Windows XP ab Service Pack 2 kann IPv6 über die Netzwerkeigenschaften hinzugefügt werden. Es gibt diverse Einschränkungen (Remote Desktop ist nicht per IPv6 erreichbar u.a.). Im grossen und ganzen funktioniert es aber.

DHCP wird allerdings nicht unterstützt, hierzu werden Hilfsmittel benötigt:

Es muss dibbler V0.8.0 heruntergeladen und installiert werden http://klub.com.pl/dhcpv6/. Bei der Installation ausschliesslich den DHCP6 Client auswählen, der Server wird nicht benötigt.

Die Konfiguration muss nicht angepasst werden, allerdings sollte der Bequemlichkeit halber dibbler als Windows Service installiert und gestartet werden. Der Rest geht automatisch.

Bestimmung der DIUD

Um die Client DIUD zu ermitteln, öffnet man im dibbler Verzeichnis (standardmässig c:\dibbler\) befindet die Datei "client-diud" mit einem Text Editor. Darin steht die ID im Klartext.

SuSE Linux Client

Für die Client-Seite muss nichts nachinstalliert werden.

Bestimmung der DIUD

Die Client DIUD ist bei SuSE ein wenig umständlicher zu ermitteln. Dazu muss man die Datei "**/var/lib/dhcp6/dhclient6.IF.lease"** öffnen, wobei IF der Name des Interfaces ist, z.B. "eth0".

Am Anfang der Datei findet sich eine Zeile wie z.B.

```
default-duid "\000\001\000\001\025\242F\215\010\000'}\366\036";
```

Dieses Beispiel entspricht der DUID 00:01:00:01:15:a2:46:8d:08:00:27:7d:f6:1e.

Um die etwas unleserliche Darstellung umzurechnen, kann man folgendes bash Script verwenden. Es akzeptiert als Parameter die ID:

```
linux-dhcp:~ # ./duidconv "\000\001\000\001\025\242F\215\010\000'}\366\036"
duid to hex converter

Input: \000\001\000\001\025\242F\215\010\000'}\366\036
Output 00:01:00:01:15:a2:46:8d:08:00:27:7d:f6:1e
```

```
#!/bin/bash
#
DUID=$1

printf "duid to hex converter\n\n"
printf "Input: %s\n"  $DUID
printf "Output "

len=1
for ((i=0; i < ${#DUID}; i+=len))
do
   if [ "${DUID:$i:1}" == "\\" ]
   then
      # 3 bytes octal
```

```
      len=4;
      oct="${DUID:$i+1:3}"
      dec=$((8#0$oct))
      printf "%02x" $dec
   else
      len=1;
      asc=${DUID:$i:1}
      dec=`printf "%c" $asc | od -A n -t u1`
      printf "%02x" $dec
   fi

   if [ $(($i+$len)) -lt ${#DUID} ]
   then
      printf ":"
   fi
done

printf "\n"
```

FreeBSD

Bei FreeBSD muss der IPv6 DHCP Client nachinstalliert werden. Der findet sich in den FreeBSD Ports KAME DHCP6 Client, Server und Relay (dhcp6-20080615_1 aus den FreeBSD Ports):

```
# cd /usr/ports/net/dhcp6
# make install
```

Weitere Details: http://www.secure-computing.net/wiki/index.php/DHCP6_Server#Installing_the_dhcp6_port_on_FreeBSD

Der Client muss nur gestartet werden, ausserdem muss in der rc.conf IPv6 noch aktiviert werden und das Interface eingestellt.

rc.conf bei FreeBSD 8.x

Folgende Zeilen müssen eingefügt bzw. angepasst werden (der Interfacename sollte mit "ifconfig" überprüft werden:

```
ipv6_enable="YES"
dhcp6c_enable="YES"
dhcp6c_interfaces="em0"
```

rc.conf bei FreeBSD 9.x

Beim neuen FreeBSD Release wurde die Unterstützung für IPv6 komplett überabeitet. Daher sehen die Einstellungen anders aus. Wichtig ist die Aktivierung des Router Advertisement Protocols, da sonst zwar IPv6 Adressen gesetzt werden. Es ist dann aber kein anderer Rechner über IPv6 erreichbar, was die Sache wohl ziemlich sinnlos macht:

```
ipv6_activate_all_interfaces="YES"
dhcp6c_enable="YES"
dhcp6c_interfaces="em0"
ifconfig_em0_ipv6="inet6 accept_rtadv"
```

Bestimmung der DIUD

Die Client DIUD steht bei FreeBSD in der Datei "**/var/db/dhcp6c_duid**". Dabei handelt es sich um eine Binärdatei, es hat also keinen Sinn, sie mit "vi" o.ä. zu öffnen.

Der Inhalt kann mit dem Befehl "**hexdump -C < /var/db/dhcp6c_duid**" leserlich gemacht werden:

```
00000000  0e 00 00 01 00 01 16 2c  15 55 08 00 27 4c 01 18  |.......,.U..'L..|
00000010
```

Alternativ geht auch "**od -t x1 < /var/db/dhcp6c_duid**":

```
0000000    0e  00  00  01  00  01  16  2c  15  55  08  00  27  4c  01  18
0000020
```

Manche Quellen (http://linux.die.net/man/8/dhcp6c) sprechen von "**/var/lib/dhcpv6/dhcp6c_duid**", im Zweifelsfall bitte in den manpages nachsehen.

Die beiden ersten Bytes geben die Länge der DIUD an, die folgenden sind die ID selber. Dabei setzt sie sich aus einer Uhrzeit (wahlweise) und der MAC Adresse zusammen, im obigen Fall war die MAC "08:00:27:4c:01:18".

Es gibt zwei Typen von DUID:

LL DUID:

```
# od -x dhcp6c_duid
0000000 000a 0300 0600 MMMM MMMM MMMM
0000014
```

LLT DUID:

```
# od -x dhcp6c_duid
0000000 000e 0100 0600 TTTT TTTT MMMM MMMM MMMM
0000020
```

"000a" stell die Länge der LL DUID von 10 in Hexadezimal und Netzwerk Byte Order dar. "000e" stellt die Länge der LLT DUID von 14 in Hexidezimal und Netzwerk Byte Order dar.

"M" ist die MAC Adresse in Host Byte Order; "T" ist die hexadezimale Zeitangabe in Host Byte Order. Die "0300", "0600", und "0100" würden als "0003", "0006" und "0001" angezeigt auf Big Endian Architekturen.

Linkliste

http://de.wikipedia.org/wiki/IPv6
http://www.daemon-systems.org/man/dhcpd.conf.5.html
DHCP6 Server V4.2.1 http://www.isc.org/software/dhcp
radvd V1.3 http://en.wikipedia.org/wiki/Radvd / http://mirrors.bieringer.de/Linux+IPv6-HOWTO-de/hints-daemons-radvd.html

SixXS - IPv6 Deployment & Tunnel Broker :: AICCU - Automatic IPv6 Connectivity Client Utility
Wikipedia - AICCU
IPv6 mit festen Adressen: So nutzt man SixXS-Tunnel | Seite 6 | Praxis | Security |

ZDNet.de
: SixXS bietet eine kostenlose IPv6-Anbindung an das Internet. ZDNet erklärt Schritt für Schritt, wie man diesen Dienst von jedem festen oder mobilen Anschluss nutzt, und wie sich eine Fritzbox oder ein Unix-Rechner als IPv6-Router einsetzen lässt.

SubnetOnline.com - Online Port Scanner IPv6

IPv6 test - API and webmaster tools
: IPv6-test.com is a free service that checks your IPv6 and IPv4 connectivity and speed. Diagnose connection problems, discover which address(es) you are currently using to browse the Internet, and what is your browser's protocol of choice when both v6 and v4 are available.

SubnetOnline.com - IPv4 to IPv6 converter
: This calculator slash converter can assit in the conversion of IPv4 type IP numbers to IPv6 notation, supporting IPv6 condensed and alternative format.

Convert IPv6 address to literal address online - ipv6-literal.net
: Online tool to convert IPv6 address to corresponding literal address. Multiple formats available. IP Calculator Shows longest and shortest IPv6 versions.

BelWue Looking Glass

Zwei Hinweise zum IPv6-Routing mit openSUSE | Marix World

Various DHCPv6 Server and Client Configuration Examples

Secure dynamic DNS howto
: how to configure secure dynamic update

IPv6 on a home network (Part 7) | Dave's Blog

ISC Dynamic Host Configuration Server (dhcpd)

Total IPv6-fähig - Homepage der Böttgers
: Die private Homepage der Familie Böttger mit Urlaubsbeschreibungen und -Bildern aus Skandinavien sowie täglichem Blog über das Leben.

DHCPv6 - Understanding of address configuration in automatic mode and installation of

DHCPv6 Server - Microsoft Windows DHCP Team Blog - Site Home - TechNet Blogs

Miredo : Introduction

IPv6 speed test - Test your IPv6 broadband speed - ipv6speedtest.net
: Test your IPv6 broadband connection speed

IPv6 Ping

SixXS - IPv6 Deployment & Tunnel Broker :: Frequently Asked Questions (FAQ)

SixXS - IPv6 Deployment & Tunnel Broker :: Tickets - Tunnel graphs show 100% packet loss, even though endpoint returns ping
: Tickets - Tunnel graphs show 100% packet loss, even though endpoint returns ping, SixXS - IPv6 Deployment and IPv6 Tunnel Broker, helping to deploy IPv6 around the world, IPv6 monitoring, IPv6 routing monitoring, IPv6 coordination, IPv6 Transition

Windows XP IPv6 - IPv6 Intelligence

DHCPv6: Dibbler - a portable DHCPv6
: DHCPv6 Linux Windows XP 2003 Dibbler

FreeBSD IPv6 - IPv6 Intelligence

SixXS Wiki - SuSE10.1 configuration

SixXS - IPv6 Deployment & Tunnel Broker :: Frequently Asked Questions (FAQ)

IPv6: Privacy Extensions einschalten | heise Netze
: Die automatische IPv6-Einrichtung nutzt auf einigen Betriebssystemen per Vorgabe die Hardware-Adresse der Netzwerkschnittstelle. Solche Adressen sind im Internet leicht wiederzuerkennen. Abhilfe schaffen die Privacy Extensions, mit denen sich

zusätzliche, über Zufallszahlen generierte und wechselnde IPv6-Adressen erstellen lassen.

USAGI Project - Linux IPv6 Development Project
La page web personnelle de Konstantin Kabassanov

DHCPv6 Configuration Examples | Michigan Tech
dhcp6c(8): DHCPv6 client daemon - Linux man page

Using DHCPv6 messages and DHCPv6 options, dhcp6c is used to request and configure IPv6 addresses and host network configuration information (i.

SourceForge.net: DHCPv6 management - xcat
radvd - Wikipedia, the free encyclopedia
IPv6 Firewalling - SixXS Wiki

Historie

Datum	Version	Änderungen
19.10.2011		Dokument erstellt
24.10.2011	1.0	Erste veröffentlichte Version
30.05.2012	1.1	FreeBSD 9.0 ergänzt
05.06.2012	1.2	Fehlende Links ergänzt, Datum V1.1 korrigiert