

IPv6 – zum Anfassen

hitforum.de

Johannes Hubertz

hubertz-it-consulting GmbH

Duisburg, 28. 4. 2010



IPv6 – zum Anfassen: Inhalt

Einleitung, Vorstellung

IPv6 – Fiktion, Hype oder nur Marketing?

Mythen und Märchen

IPv4 – Header – IPv6

IPv6 – genug Adressen, nie und nimmer NAT

IPv6 – next header – IPv6

ICMPv6 – Das Wichtigste

IPv6 Autokonfiguration

IPv6 Konfiguration in Linux, OpenBSD und XP (M\$)

IPv6 – Paketfilter

Transportvehikel: IPv6 über IPv4

Quellen und Lesetipps



Vorstellung: Johannes Hubertz

- 1980 Hardware-Reparatur: Ersatzteile auf Bauteilebene
- 1984 Entwicklung Sonderprodukte, Assembler, PLM
- 1987 Erstkontakt mit Unix (SCO-Xenix) und C
- 1994 Erstkontakt mit IP
- 1996 Xlink, root@www.bundestag.de, ...
- 1997 SSLeay, ipfwadm mit shell-scripts
- 1998 „Ins Allerheiligste“, iX 1/1998, Heise Verlag
- 1999 IT-Security Mgr. Bull D-A-CH
- 2001 Gibraltar, FreeSwan, iptables ...
- 2001 Firmenteilung → Entwicklung und Betrieb sspe
- 2002 Weiterentwicklung und Betrieb von sspe
- 2005 Gründung der hubertz-it-consulting GmbH
- ab 1973 Bundesanstalt Technisches Hilfswerk in Köln-Porz, ... ZFü, S2
- ab 2001 Segeln, am liebsten auf Salzwasser, SKS



Vorstellung: hubertz-it-consulting GmbH

Erkenntnisse aus dem Berufsleben

Bellovin and Cheswick: Firewalls and Internet Security, 1994

Fazit: Keep it simple!

Oder mit Einstein: So einfach wie möglich, aber nicht einfacher!

Etwas Erfahrung war Voraussetzung

Gründung am 8. August 2005, Sitz in Köln

Geschäftsinhalt: Dienstleistungen im Umfeld der IT-Sicherheit

Logo: Johannes Hubertz Certificate Authority als ASCII-7Bitmuster

Diese paar Bits findet sich in einigen 10000 X.509 Anwenderzertifikaten in der Seriennummer wieder

Wir sind käuflich ;-)



IPv6 – nur Marketing?



Haben Sie eine Idee,

wie Ihr Mobiltelefon

mit Ihrem PC

via bluetooth . . .

???



**Ich denke nie an die Zukunft,
sie kommt früh genug.**

Albert Einstein



RIPE Community resolution: (27. October 2007)

„Growth and innovation on the Internet depends on the continued availability of IP address space. The remaining pool of unallocated IPv4 address space is likely to be fully allocated within two to four years. IPv6 provides the necessary address space for future growth. We therefore need to facilitate the wider deployment of IPv6 addresses“



IPv6 – Fiktion, Hype, Marketing, oder was?

IPv6 ist schon betagt:

Start in der Mitte der 90er Jahre

IPv6 ist reine Technik:

weit mehr als 200 RFCs spezifizieren IPv6

IPv6 ist notwendig:

IPv4 wird knapp

IPv6 existiert:

IPv6 ist fertig, funktioniert schon bestens und bald omnipräsent



ewige Wahrheiten . . .

**Menschen mit einer neuen Idee gelten solange als Spinner,
bis sich die Sache durchgesetzt hat.**

Mark Twain



Mythen und Märchen



Die schlechte Nachricht

IPv6 ist genauso unsicher wie IPv4

really!



Die gute Nachricht

IPv6 ist genauso sicher wie IPv4

really!



ICMPv6 ist böse

ICMPv6 ist essentieller Bestandteil von IPv6.

ICMPv6 komplett zu filtern (wie es bei IPv4 oft üblich war), ist gleichbedeutend mit IPv6 ausschalten. Es funktioniert nicht.



IPsec ist schon drin!

IPsec wurde zuerst auf IPv6 spezifiziert

Jede IPv6-Implementierung sollte auch IPsec realisieren. Jedoch:
Zur Nutzung braucht man Schlüsselverteilung,
z. B. IKEv1 oder IKEv2 oder proprietär, Preshared Keys oder X.509-Zertifikate
KnowHow-trächtige Konfigurationen sind gefragt,
Kompatibilität unter Herstellern,
... nix ändert sich, was reg ich mich auf ...;-)

Wir brauchen nur eine

PKI!

Dann ist die Welt ja in Ordnung.



IPv4 – IPv6 unvergleichbar, oder?



IP – header as defined in RFC 791 as of September 1981

ASCII – ART

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
!Version!  IHL  !Type of Service!          Total Length  !
+-----+-----+-----+-----+-----+-----+-----+-----+
!          Identification          !Flags!      Fragment Offset  !
+-----+-----+-----+-----+-----+-----+-----+-----+
! Time to Live !      Protocol  !          Header Checksum  !
+-----+-----+-----+-----+-----+-----+-----+-----+
!          Source Address          !
+-----+-----+-----+-----+-----+-----+-----+-----+
!          Destination Address    !
+-----+-----+-----+-----+-----+-----+-----+-----+
!          Options                !      Padding  !
+-----+-----+-----+-----+-----+-----+-----+-----+
```



IPv6 – header as defined in RFC 2460 as of December 1998 obsoletes RFC 1883 (December 1995)

ASCII – ART

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|Version|          Flow Label  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Payload Length      | Next Header |      Hop Limit  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Source Address 128 bits          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Destination Address 128 bits          |
+-----+-----+-----+-----+-----+-----+-----+-----+
```



IPv6 – header explanations

Version	4 Bit	IP Version (==6)
Flowlabel	28 Bit	Zusatzinformationen für Router, z.B. für QOS
Payload Length	16 Bit	Länge des Paketes nach dem Header
Next Header	8 Bit	Welcher Header kommt danach?
Hop Limit	8 Bit	vgl. TTL bei IPv4
Source Address	128 Bit	
Destination Address	128 Bit	



IPv6 – address explained . . .

128 Bit are combined from:

bits number	value example	meaning
3	001 _{bin} 111 _{bin}	prefix global allocatable 2000:: 3<br/ multicasts et.al.
45	2001:db8:: 32<br/ 2001:db8:f002:: 48</td <td>global routing prefix documentation RIPE, ISP, customer friendly user test</td>	global routing prefix documentation RIPE, ISP, customer friendly user test
16	0001 _{hex} ... ffff _{hex}	subnet ID second usable subnet /64 another $2^{16} - 3$ of /64 last usable /64
64	216:d3ff:fea4:5174	Interface ID a Laptop's ethernet interface ID

Es gibt keine Broadcasts mehr!



Adressen – Kleinigkeiten!



IPv6 – Adresseigenschaften

IPv6 Adressen sind 128 bit lang, d.h. $IP \in \{1..2^{128}\}$

$2^{128} = 340.282.366.920.938.463.463.374.607.431.768.211.456_{dez}$

Das entspricht 665 Milliarden Adressen pro mm^2 Erdoberfläche

Hexadezimale Schreibweise, je 2 Bytes durch einen ':' getrennt.

Beispiel: 2001:db8:f002:4711::1 '::' \Leftrightarrow fehlende 0-Bytes

'::' Nur **einmal** in einer Adresse!

2001:db8:f002::/48 \Leftrightarrow 65.536 /64 Netze (prefix wie bei IPv4)



IPv6 – Scope (Reichweite)

link-local

Jedes Interface hat 1 . . n link-local Adresse(n) aus fe80::/10

site-local deprecated!

Ein Interface hat 0 . . n site-local Adressen aus fc00::/8 oder fd00::/8

global

Ein Interface hat 0 . . n globale Adressen aus 2000::/3

multicast

Ein Interface hat 0 . . n Multicast Adressen aus ff00::/8



IPv6 – Besondere Adressen

::	nicht spezifizierte Adresse \Leftrightarrow 0:0:0:0:0:0:0:0
::1	loopback
fe80::/10	link-local
ff00::/8	multicast
ff01::1	multicast, all hosts
ff01::2	multicast, all routers
fc00::/8	Unique Local Adressen (zentral verwaltet)
fd00::/8	Unique Local Adressen
2000::/3	globale Unicast Adressen
2001:db8::/32	Prefix für Dokumentation



next header

Beliebig viele Header pro IP-Paket



IPv6 header: next header I

IPv6 header: next header field, 8 bits

- ähnlich dem Protocol-Feld im IPv4-Header, aber universeller
- Werte gleich, siehe auch `/etc/protocols`
- beliebig verkettbar
- RH Typ 0 und 2 bergen einige Sicherheitsrisiken
- RH Typ 0 ist „deprecated“ \Rightarrow , ist aber schon implementiert!(RFC 5095)
- RH Typ 2 ist essentieller Bestandteil von Mobile IPv6

Lösungsansätze gibt es, aber

- entweder kompliziert, z.B. im Firewalling
- oder noch nicht allgemein implementiert (RFC 5095)



IPv6 header: next header II

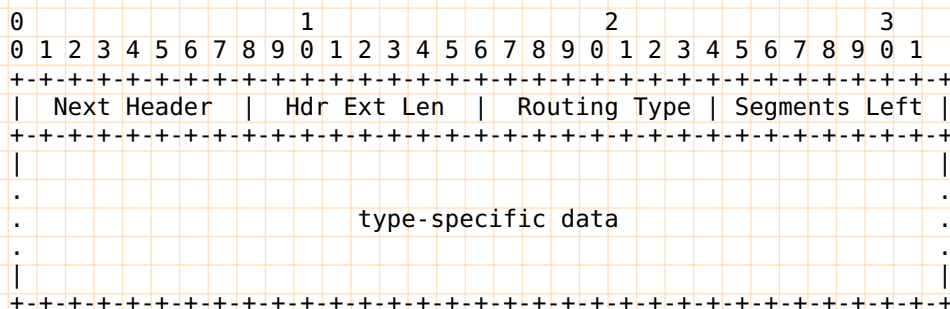
Extension Header in der Basis-Spezifikation

value	meaning
0	hop by hop options header
43	routing header
44	fragmentation header
50	encapsulation payload header (RFC 2406)
51	authentication header (RFC 2402)
60	destination options header



IPv6 header: next header III

routing header *müssen* von jedem Host verarbeitet werden



routing header type 0

Entspricht dem (strict/loose) sourcerouting, für debug, ISP-spez.Routing, ...

routing header type 2

Wird zwingend für Mobile-IPv6 benötigt



IPv6 header: next header IV

Verkettungen sind in beliebiger Länge möglich:

IPv6 header Next header = TCP value 6	TCP header and data		
IPv6 header Next header = Routing Value 43	Routing header Next header = TCP Value = 6	TCP header and data	
IPv6 header Next header = Routing Value 43	Routing header Next header = Fragment Value = 44	Fragment header Next header = TCP Value 6	TCP header and data



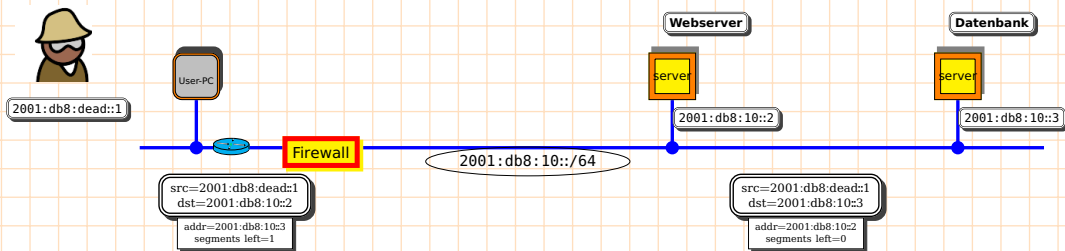
IPv6 header: next header V

Reihenfolge der extension header:

1	IPv6 Header
2	Hop-by-Hop Options Header
3	Destination Options Header für Router auf dem Pfad
4	Routing Header
5	Fragment Header
6	Authentication Header
7	Encapsulation Security Payload Header
8	Destination Options Header für den endgültigen Empfänger
9	Upper-Layer Header



next-header – malicious usage



Angrifer schickt ein Paket an die erreichbare Adresse ::2, dieser Host leitet es weiter an ::3, welcher durch Firewall strikt gefiltert wird.



ICMPv6

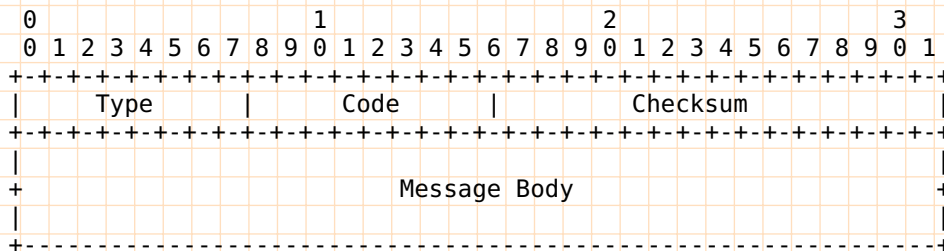
internet control message protocol version 6



ICMPv6 – as defined in RFC 2463

ICMPv6

- 1.) ersetzt ARP vollständig: **neighbor discovery (ND)**
- 2.) kann automatisch Routen: **router discovery (RD)**
- 3.) erkennt doppelte Adressen: **duplicate address detection (DAD)**
- 4.) nutzt multicasts: **ff00::0 ip6-mcastprefix**



ICMPv6 ist essentieller Bestandteil der Ende-zu-Ende Kommunikation!
Daher: Filterung von ICMPv6 **nur** auf **spezielle icmp-types** möglich



ICMPv6 – as defined in RFC 2463

type	meaning
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect
138	Router Renumbering



ND – neighbor detection

neighbour detection

ND ersetzt arp (address resolution protocol) RFC 2461, December 1998:

„Neighbor Solicitation: Sent by a node to determine the link-layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link-layer address. Neighbor Solicitations are also used for Duplicate Address Detection.

Neighbor Advertisement: A response to a Neighbor Solicitation message. A node may also send unsolicited Neighbor Advertisements to announce a link-layer address change.

- 1 prefix ff02::1:ff00:0/104 und rechte 3 Oktetts der Ziel-IP → Multicast-Adresse
- 2 icmpv6 type 135 an diese Adresse
- 3 Zielhost antwortet mit icmpv6 type 136 oder timeout



DAD – duplicated address detection

duplicate address detection

DAD geschieht vor der Zuweisung der eigenen IPv6-Adresse, RFC 2462, December 1998:

„Duplicate Address Detection is performed on unicast addresses prior to assigning them to an interface whose DupAddrDetectTransmits variable is greater than zero. Duplicate Address Detection MUST take place on all unicast addresses, regardless of whether they are obtained through stateful, stateless or manual configuration, with the exception of the following ... “

- 1 Unicast, ICMP Typ 135, Absender ':::' an die Zieladresse
- 2 Falls vorhanden, erfolgt eine Antwort an ff02::1
- 3 Falls nicht vorhanden, wird die Adresse gewählt



IPv6 – autoconfiguration



automagically configuration of workstations. . .

IPv6 – fast alles kann automatisch geschehen . . .

- ① Interface-ID (MAC-Adresse) bestimmt link-local-Adresse → fe00::/64
- ② DAD = duplicate address detection
- ③ RD = router detection
- ④ aktiver Host am LAN, erkennt Routing-Änderungen mit RD



Linux

interfaces – configuration



IPv6 Interface: Linux configuration

static:

```
tut:~# more /etc/network/interfaces 1
# The loopback network interface 2
auto lo 3
iface lo inet loopback 4
# 5
auto eth2 6
iface eth2 inet6 static 7
    pre-up /sbin/modprobe ipv6 8
    address 2001:db8:f002:fb00::1 9
    netmask 64 10
    post-up /sbin/ip route add 2001:db8:f002:fb00::/56 via 2001:db8:f002:fb00::2 11
    post-up echo "1" > /proc/sys/net/ipv6/conf/all/forwarding 12
```

autoconfiguration:

```
tut:~# more /etc/network/interfaces 1
# The loopback network interface 2
auto lo 3
iface lo inet loopback 4
# The primary network interface 5
allow-hotplug eth0 6
iface eth0 inet6 manual 7
    pre-up /sbin/ip link set dev eth0 up 8
    post-down /sbin/ip link set dev eth0 down 9
```



IPv6 Linux commands

<code>ip -6 address add nnn dev eth0 scope global</code>	IP-Adresse hinzufuegen
<code>ip -6 address del nnn dev eth0</code>	IP-Adresse wegnehmen
<code>ip -6 neigh show</code>	Nachbarschaft anzeigen
<code>ip -6 route show</code>	IPv6 Routen auflisten
<code>ip -6 route add target via nexthop dev eth0</code>	IPv6 Route hinzufuegen
<code>ip -6 route del target via nexthop dev eth0</code>	IPv6 Route wegnehmen



IPv6 Interface: OpenBSD configuration

static:

```
# cat hostname.sis0
inet 192.168.110.177 255.255.255.0 NONE
inet6 2001:db8:f002:2::10/64
#
# grep rtadvd rc.conf
rtadvd_flags=NO # for normal use: list of interfaces
#
```

autoconfiguration:

```
# grep ip6 sysctl.conf
net.inet6.ip6.forwarding=0 # 1=Permit forwarding (routing) of IPv6 packets
#net.inet6.ip6.mforwarding=1 # 1=Permit forwarding (routing) of IPv6 multicast packets
#net.inet6.ip6.multipath=1 # 1=Enable IPv6 multipath routing
net.inet6.ip6.accept_rtadv=1 # 1=Permit IPv6 autoconf (forwarding must be 0)
#
```



IPv6 OpenBSD commands

```
ifconfig      Alle Interfaces anzeigen [Mix aus L1,L2,L3]
ndp -an       Nachbarschaft anzeigen
route -n show  Routingtabelle anzeigen
```



IPv6 Interface: M\$ XP configuration

static:

AFAIK: NA

autoconfiguration:

```
cmd.exe 1
ipv6.exe install 2
```

reboot (as usual) oder: Zum Beenden drücken Sie **Start!**



IPv6 M\$ XP commands

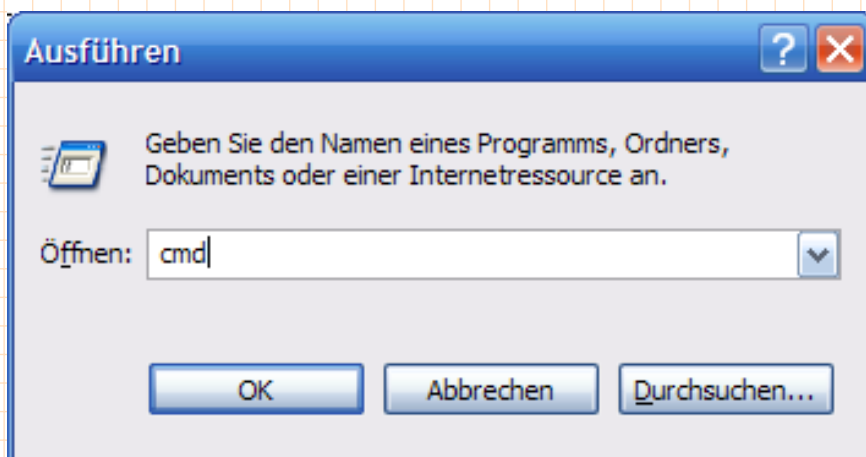
<code>ipconfig /all</code>	IP Konfiguration anzeigen
<code>route print</code>	Routingtabelle anzeigen

Die folgenden Bilder sagen mehr als viele Worte!



Da geht nix automatisch!

Zum Beenden drücken Sie erst auf Start¹, dann auf ausführen,
„cmd“ eingeben, dann OK klicken!



¹Links unten, oder woanders



Da geht immer noch nix automatisch!

```
cmd.exe - cmd
C:\Dokumente und Einstellungen>ip6
Syntax: ip6 [-p] [-v] if [Schnittstellenindex]
ip6 [-p] ifcr v6v4 v4src v4dst [nd] [pml]
ip6 [-p] ifcr 6over4 v4src
ip6 [-p] ifc Schnittstellenindex [forwards] [-forwards] [advertises] [-
advertises] [mtu Anz. Bytes] [site Sitekennung] [preference P]
ip6 rlu Schnittstellenindex v4dst
ip6 [-p] ifd Schnittstellenindex
ip6 [-p] adu Schnittstellenindex/Adresse [life validlifetime[/preflifet
ime]] [anycast] [unicast]
ip6 nc [Schnittstellenindex [Adresse]]
ip6 ncf [Schnittstellenindex [Adresse]]
ip6 rc [Schnittstellenindex Adresse]
ip6 rcf [Schnittstellenindex [Adresse]]
ip6 hc
ip6 [-p] [-v] rt
ip6 [-p] rtu Präfix Schnittstellenindex[/Adresse] [life valid[/pref]] [
preference P] [publish] [alter] [spl Sitepräfixlänge]
ip6 spt
ip6 spu Präfix Schnittstellenindex [life L]
ip6 [-p] sp
ip6 [-p] gpu [Parameter Wert] ... (eventuell -?)
ip6 renew [Schnittstellenindex]
ip6 [-p] ppt
ip6 [-p] ppu Präfix Reihenfolge P srclabel SL [dstlabel DL]
ip6 [-p] ppd Präfix
ip6 [-p] reset
ip6 install
ip6 uninstall
Einige Unterbefehle erfordern lokale Administratorrechte.
C:\Dokumente und Einstellungen>
```

Hier ist nix zum klicken, erst mal nur ansehen!
Alles klar?



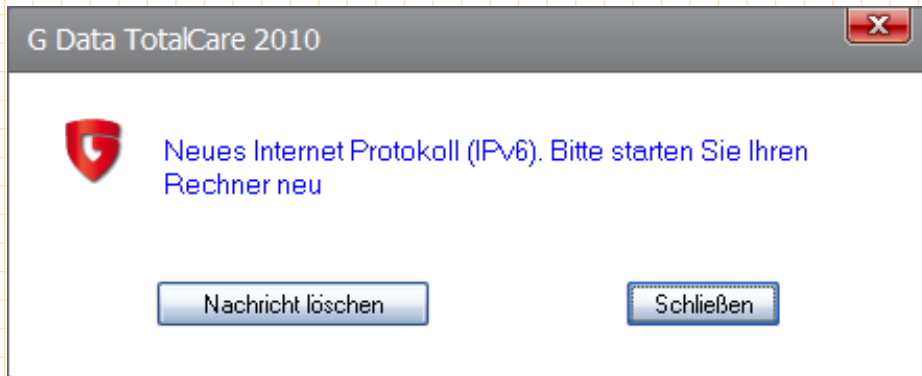
Und immer noch nix automatisch!

```
cmd.exe - cmd
C:\Dokumente und Einstellungen>ip6 install
Installation wird durchgeführt...
Erfolgreich.
C:\Dokumente und Einstellungen>_
```

Gefühlte Wartezeit: 1 min,
ansonsten bleibt fast alles unsichtbar. Hm.



Wer behindert mich hier?



Oh, welche Nachricht darf / kann / soll / muß ich löschen?



Nur ein Boot später . . .

. . . start, ausführen, cmd, Sie wissen schon . . .

```
cmd Eingabeaufforderung
C:\Dokumente und Einstellungen>ipconfig
Windows-IP-Konfiguration

Ethernetadapter LAN-Verbindung:

    Verbindungsspezifisches DNS-Suffix: hubertz.de
    IP-Adresse. . . . . : 192.168.110.225
    Subnetzmaske. . . . . : 255.255.255.0
    IP-Adresse. . . . . : 2001:4dd0:f002:2:156f:f4b2:25:305f
    IP-Adresse. . . . . : 2001:4dd0:f002:2:240:d0ff:fe89:d30f
    IP-Adresse. . . . . : fe80::240:d0ff:fe89:d30f%4
    Standardgateway . . . . . : 192.168.110.254
    fe80::200:24ff:fec8:cf04%4

Tunneladapter Teredo Tunneling Pseudo-Interface:

    Verbindungsspezifisches DNS-Suffix:
    IP-Adresse. . . . . : fe80::ffff:ffff:ffff%5
    Standardgateway . . . . . :

Tunneladapter Automatic Tunneling Pseudo-Interface:

    Verbindungsspezifisches DNS-Suffix: hubertz.de
    IP-Adresse. . . . . : fe80::5efe:192.168.110.225%2
    Standardgateway . . . . . :

C:\Dokumente und Einstellungen>_
```

Das freut ET: Nach Hause telefonieren ist auch dabei!
Teredo sei Dank!



Nur ein Boot später ...

```
C:\Dokumente und Einstellungen>ping6 ipv6.google.com
ipv6.l.google.com [2a00:1450:8006::69] wird angepingt
von 2001:4dd0:f002:2:156f:f4b2:25:305f mit 32 Bytes Daten:
Antwort von 2a00:1450:8006::69: Bytes=32 Zeit=41ms
Antwort von 2a00:1450:8006::69: Bytes=32 Zeit=40ms
Antwort von 2a00:1450:8006::69: Bytes=32 Zeit=38ms
Antwort von 2a00:1450:8006::69: Bytes=32 Zeit=40ms

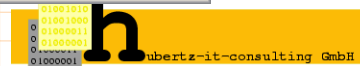
Ping-Statistik für 2a00:1450:8006::69
Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
Ungefähre Zeitangaben in Millisekunden:
Minimum = 38ms, Maximum = 41ms, Mittelwert = 39ms

C:\Dokumente und Einstellungen>tracert6 f.nic.de
Routenverfolgung zu f.nic.de [2001:608:6:6::10]
von 2001:4dd0:f002:2:156f:f4b2:25:305f über eine maximale Anzahl von 30 Hops:

 1          1 Msek      1 Msek      1 Msek      2001:4dd0:f002:2::1
 2          23 Msek     24 Msek     25 Msek     2001:4dd0:f002:1::2
 3          25 Msek     25 Msek     23 Msek     2001:4dd0:f002:1::53
 4          25 Msek     24 Msek     26 Msek     2001:4dd0:f002::1
 5          31 Msek     32 Msek     105 Msek    2002:c30e:f747:624::1
 6          33 Msek     31 Msek     33 Msek     2001:4dd0:a2b:1:dc30::c
 7          30 Msek     30 Msek     32 Msek     2001:4dd0:a2b:11:dc30::2
 8          34 Msek     32 Msek     33 Msek     2001:4dd0:a2b:14:10::b
 9          35 Msek     39 Msek     35 Msek     Cisco-F-UI-Gi2-5.Space.net [2001:7f8::1]
5a3:0:11
10         36 Msek     36 Msek     35 Msek     denic-router-2.denic.de [2001:608:0:f01
::921
11         36 Msek     34 Msek     37 Msek     2001:608:6:6::10

Ablaufverfolgung beendet.
C:\Dokumente und Einstellungen>
```

Da geht schon mal was. Aber: Msek oder ms oder wie?



Connectivity ...

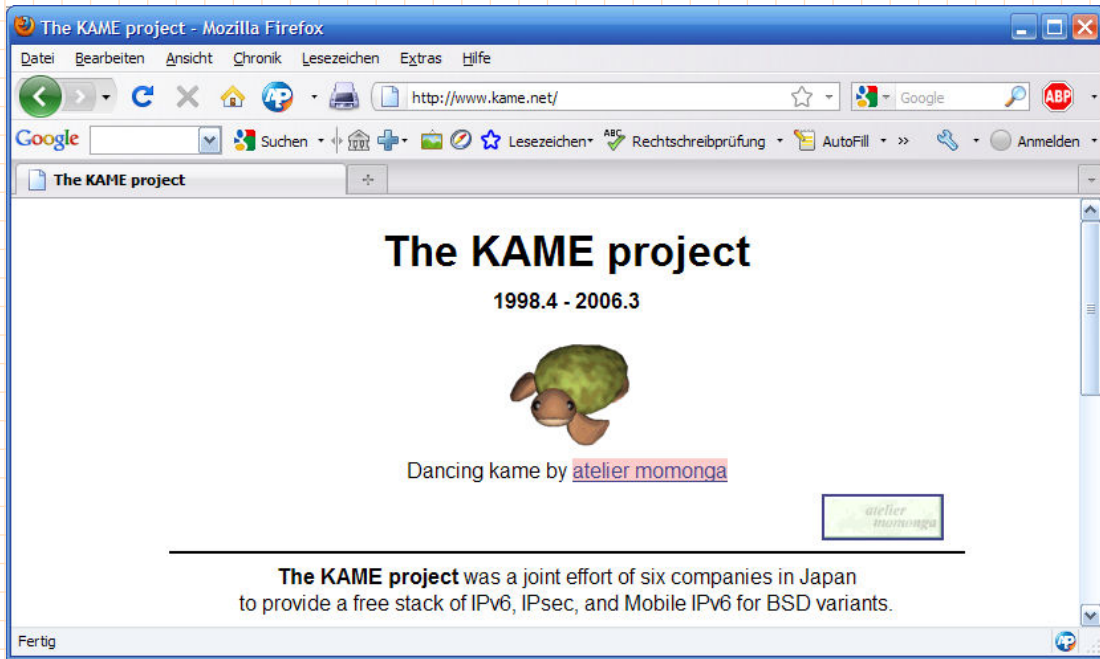
```
C:\Dokumente und Einstellungen>ipv6 -v rt
fe80::5efe:192.168.110.225/128 -> 2/fe80::5efe:192.168.110.225 Präferenz 1if+4=5
Gültigkeitsdauer infinite (System)
2001:4dd0:f002:2:156f:f4b2:25:305f/128 -> 4/2001:4dd0:f002:2:156f:f4b2:25:305f P
räferenz 4 Gültigkeitsdauer infinite (System)
2001:4dd0:f002:2:240:d0ff:fe89:d30f/128 -> 4/2001:4dd0:f002:2:240:d0ff:fe89:d30f
Präferenz 4 Gültigkeitsdauer infinite (System)
2001:4dd0:f002:2::/64 -> 4 Präferenz 8 Gültigkeitsdauer 2591704s (Autokonf)
::/0 -> 4/fe80::200:24ff:fec8:cf04 Präferenz 256 Gültigkeitsdauer 1504s (Autokon
f)
fe80::ffff:ffff:ffffd/128 -> 5/fe80::ffff:ffff:ffffd Präferenz 2if+4=6 Gültigkeits
dauer infinite (System)
ff00::/8 -> 4 Präferenz 8 Gültigkeitsdauer infinite (System)
fe80::240:d0ff:fe89:d30f/128 -> 4/fe80::240:d0ff:fe89:d30f Präferenz 4 Gültigkei
tsdauer infinite (System)
::1/128 -> 1/::1 Präferenz 4 Gültigkeitsdauer infinite (System)
ff00::/8 -> 1 Präferenz 8 Gültigkeitsdauer infinite (System)
fe80::1/128 -> 1/fe80::1 Präferenz 4 Gültigkeitsdauer infinite (System)

C:\Dokumente und Einstellungen>_
```

Bisserl unleserlich, aber vermutl. korrekt!



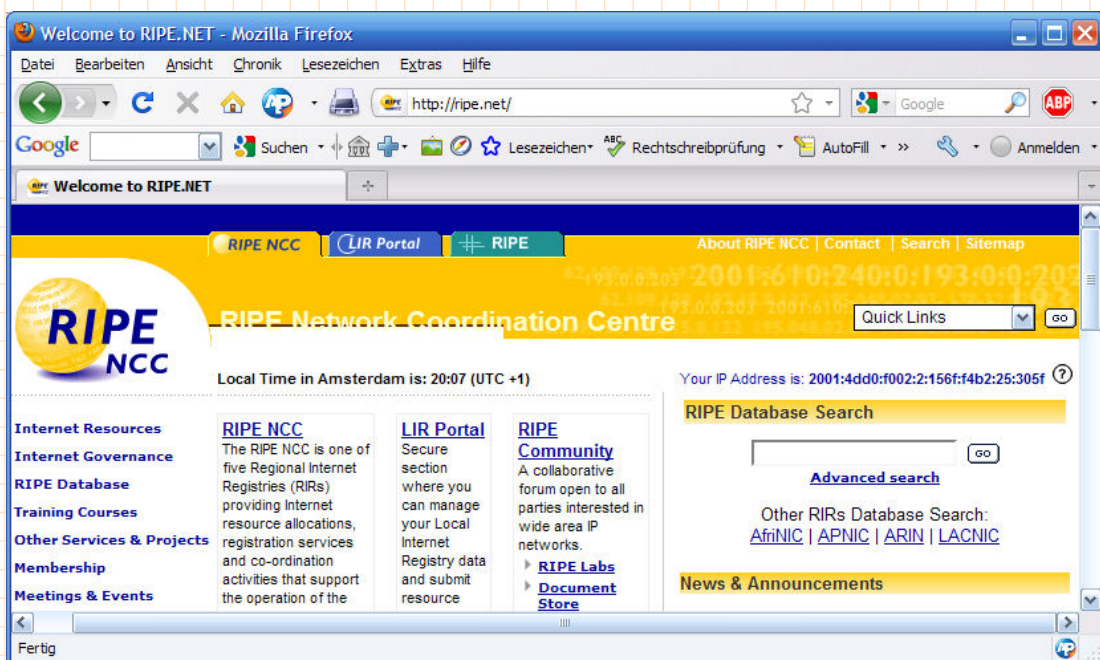
Connectivity ...



Die rennt aber schnell für eine Schildkröte (IPv6)!



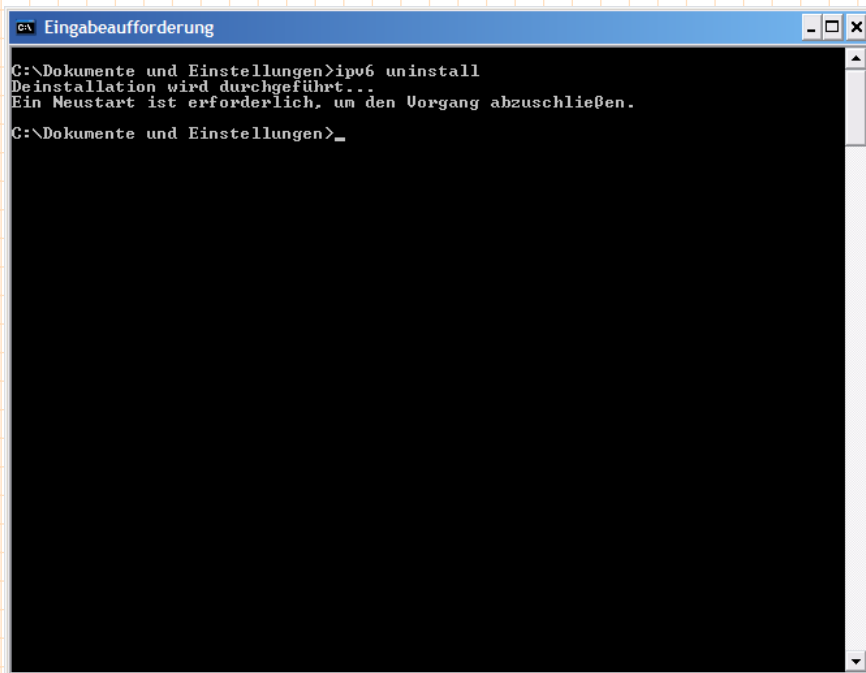
Connectivity ...



Wer bin ich?

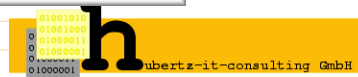


...jetzt aber genug!



```
C:\Dokumente und Einstellungen>ipv6 uninstall
Deinstallation wird durchgeführt...
Ein Neustart ist erforderlich, um den Vorgang abzuschließen.
C:\Dokumente und Einstellungen>_
```

Gefühlte Wartezeit: Wieder ca. 1 min,
hier bleibt alles unsichtbar. Hm.



Client Autoconfiguration ...

Server and clients view



IPv6 clients autoconfiguration (magic)

radvd server-config-file² follows:

```
1 interface eth2 {
2   AdvSendAdvert on;
3   prefix 2001:db8:f002:fc00::/64   { };
4 };
```

rtadvd server-config-file³ follows:

```
1 # cat /etc/rtadvd.conf
2 sis1: \
3   prefix: 2001:db8:f002::/48
4   prefix: 2001:db8:f002:0020::/64
5 #
```

Remark: radvd – router advertisement daemon

²Debian GNU/Linux: /etc/radvd.conf

³OpenBSD: /etc/rtadvd.conf



IPv6 clients autoconfiguration (magic)

dibbler server-config-file⁴ follows:

```
1 log-level 8
2 log-mode short
3 preference 0
4 iface "eth2" {
5   // also ranges can be defines, instead of exact values
6   t1 1800-2000
7   t2 2700-3000
8   preferred-lifetime 3600
9   valid-lifetime 7200
10  class {
11    pool 2001:db8:f002:fc00:0100:0000::/96
12  }
13  ta-class {
14    pool 2001:db8:f002:fc00:0200:0000::/96
15  }
16  pd-class {
17    pd-pool 2001:db8:f002:fc00:0300::/80
18    pd-length 96
19  }
20 option dns-server 2001:db8:f002:1::53
21 option domain hubertz.de
22 option vendor-spec 5678-0x0002aaaa
23 option ntp-server 2001:db8:f002:1::1
24 option time-zone CET
25 }
```

Remark: dibbler-server is a portable implementation of the DHCPv6 server.

⁴Debian GNU/Linux: /etc/dibbler/server.conf



IPv6 client autoconfiguration (magic only)

dibbler client-config-file⁵ follows:

```
1 # 8 (Debug) is most verbose. 7 (Info) is usually the best option
2 log-level 7
3 # To perform stateless (i.e. options only) configuration, uncomment
4 # this line below and remove any "ia" keywords from interface definitions
5 #
6 # stateless
7 #
8 iface eth0 {
9 # ask for address
10     ia
11 # ask for options
12     option dns-server
13     option domain
14 # option ntp-server
15 # option time-zone
16 # option sip-server
17 # option sip-domain
18 # option nis-server
19 # option nis-domain
20 # option nis+-server
21 # option nis+-domain
22 }
```

Remark: dibbler-client is a portable implementation of the DHCPv6 client.

⁵Debian GNU/Linux: /etc/dibbler/client.conf



Alle Clients sind konfiguriert,

Was nun?



Wir haben eine Firewall!

Da ist alles sicher.

?



NetFilter

ip6tables, pf – und wie?



NetFilter – linux 2.4, 2.6, ...

altbewährte Technik mit 3 Standard-chains

- INPUT
- OUTPUT
- FORWARD
- userdefined chains

Ein Kommando für alles

`man ip6tables`

`/sbin/ip6tables --help`



`/sbin/ip6tables` – linux 2.4, 2.6, ...

```
1 #
2 # /sbin/ip6tables --help
3
4 Usage: ip6tables -[AD] chain rule-specification [options]
5         ip6tables -I chain [rulenum] rule-specification [options]
6         ip6tables -R chain rulenum rule-specification [options]
7         ip6tables -D chain rulenum [options]
8         ip6tables -[LS] [chain [rulenum]] [options]
9         ip6tables -[FZ] [chain] [options]
10        ip6tables -[NX] chain
11        ip6tables -E old-chain-name new-chain-name
12        ip6tables -P chain target [options]
13        ip6tables -h (print this help information)
14
15 . . .
```

Die man-page liest sich noch mühsamer!

Der Mensch ist faul und infolgedessen auch erfindungsreich ...



Paketfilter I: Hier ist Handarbeit gefragt

Definitionen: Wer ist hier beteiligt?

#Name	Adresse	Kommentar
any	::/0	# Alle Welt
many	2000::/3	# erreichbare Welt
localhost	::1/128	#
srv	2001:db8:f002:2::10/128	# service
ns	2001:db8:f002:1::53/128	# 1.nameserver
ns	2001:db8:f002:3::23/128	# 2.nameserver
nc-dns	2001:4dd9:abcd:0:0:2:b351:f602/128	# NetCologne DNS
admin	fe80::a00:27ff:fe67:4649/128	# administration
admin	2001:db8:f002:3:a00:27ff:fe67:4649/128	# administration

Regeln: Wer darf mit wem was?

#Source	Destination	protocol	port	action	Options	#Kommentar
admin	ns	tcp	22	accept	NOIF	
many	ns	udp	53	accept		
any	any	ip6	all	drop		# no def. log



Paketfilter II: Handarbeit nicht mehr so sehr gefragt

Generierung von Paket-Filtern für Linux und OpenBSD

Output Stufe I Linux/OpenBSD

```
!mach!rn!me_s!me_d!src!src_if!dst!dst_if!prot!port!acti!varying options!  
!ns!1!0!0!2001:db8:f002:3:a00:27ff:fe67:4649/128!!2001:db8:f002:1::53/128!eth1!tcp!22!accept!NOIF!  
!ns!1!0!1!2001:db8:f002:3:a00:27ff:fe67:4649/128!!2001:db8:f002:3::23/128!eth1!tcp!22!accept!NOIF!  
!ns!2!0!0!2000::/3!!2001:db8:f002:1::53/128!eth1!udp!53!accept!!  
!ns!2!0!1!2000::/3!!2001:db8:f002:3::23/128!eth1!udp!53!accept!!  
!ns!3!0!0!::/0!!::/0!!ip6!all!drop!!
```

Output Stufe II Linux

```
export IPI="/sbin/ip6tables -A INPUT "  
export IPO="/sbin/ip6tables -A OUTPUT "  
export IPF="/sbin/ip6tables -A FORWARD "  
$IPI -m rt -rt-type 0 -j DROP  
$IPF -m rt -rt-type 0 -j DROP  
echo "Rule 1 \r">2  
$IPF -s 2001:db8:f002:3:a00:27ff:fe67:4649/128 -d 2001:db8:f002:1::53/128 -p tcp -dport 22 -j ACCEPT  
$IPF -d 2001:db8:f002:3:a00:27ff:fe67:4649/128 -s 2001:db8:f002:1::53/128 -p tcp -sport 22 -j ACCEPT  
$IPI -s 2001:db8:f002:3:a00:27ff:fe67:4649/128 -d 2001:db8:f002:3::23/128 -p tcp -dport 22 -j ACCEPT  
$IPO -d 2001:db8:f002:3:a00:27ff:fe67:4649/128 -s 2001:db8:f002:3::23/128 -p tcp -sport 22 -j ACCEPT  
echo "Rule 2 \r">2  
...  
/bin/ip6tables -P INPUT DROP  
/bin/ip6tables -P OUTPUT DROP  
/bin/ip6tables -P FORWARD DROP
```



IPv6 – transportation over IPv4



IPv6 transportation via IPv4

OpenVPN is our friend, server-config-file⁶ follows:

```
1 daemon dns
2 local 192.0.2.126 # Test-Net of RFC 3330 - Special-Use IPv4-Addresses
3 port 4712
4 dev tun
5 tun-ipv6
6 proto tcp-server
7 ifconfig-noexec
8 redirect-gateway
9 up /etc/openvpn/up-c
10 down /etc/openvpn/down-c
11 log-append /var/log/ovpn-c.log
12 writepid /var/run/ovpn-c.pid
13 dh /etc/openvpn/dh2048.pem
14 ca /etc/openvpn/cacert.pem
15 cert /etc/openvpn/r-exCert.pem
16 key /etc/openvpn/r-exReq.pem
17 crl-verify /etc/openvpn/crl.pem
18 tls-server
19 tls-verify /etc/openvpn/tls-verify
20 tls-exit
21 persist-tun
22 ping 15
23 ping-restart 45
24 ping-timer-rem
25 persist-key
26 verb 3
```

Remark: one server-instance for each possible client connection

⁶Debian GNU/Linux



IPv6 transportation via IPv4

OpenVPN is our friend, server-config-script up-c follows:

```
1 #!/bin/bash
2 #
3 INTERFACE=$1; shift;
4 TUN_MTU=$1; shift;
5 UDP_MTU=$1; shift;
6 LOCAL_IP=$1; shift;
7 REMOTE_IP=$1; shift;
8 MODUS=$1; shift;
9 #
10 ip link set ${INTERFACE} up
11 ip link set mtu ${TUN_MTU} dev ${INTERFACE}
12 #
13 for myad in fc00
14 do
15     ip -6 addr add fe80:0:${myad}::1/64 dev ${INTERFACE}
16     ip -6 route add 2001:db8:f002:${myad}::/56 via fe80:0:${myad}::2 dev ${INTERFACE}
17 done
18 exit 0
```

Remark: one server-instance for each possible client connection



IPv6 transportation via IPv4

OpenVPN is our friend, serverside tls-verify-script follows:

```
1 #!/usr/bin/perl -w
2 #
3 ($depth, $cn) = @ARGV;
4 if ($depth == 1 && $cn eq '/C=DE/ST=Germany/L=Cologne/O=hubertz-it-consulting_GmbH/OU=consulting/CN=
5     hubertz_consulting_CA/emailAddress=ca@hubertz.de') {
6     exit 0;
7 }
8 exit 1 if (!$depth == 0);
9 exit 0 if ($cn eq '/C=DE/ST=Germany/O=hubertz-it-consulting_GmbH/OU=IPv6-Tunnel/CN=fa00');
10 exit 0 if ($cn eq '/C=DE/ST=Germany/O=hubertz-it-consulting_GmbH/OU=IPv6-Tunnel/CN=fb00');
11 exit 0 if ($cn eq '/C=DE/ST=Germany/O=hubertz-it-consulting_GmbH/OU=IPv6-Tunnel/CN=fc00');
12 exit 0 if ($cn eq '/C=DE/ST=Germany/O=hubertz-it-consulting_GmbH/OU=IPv6-Tunnel/CN=fd00');
13 exit 0 if ($cn eq '/C=DE/ST=Germany/O=hubertz-it-consulting_GmbH/OU=IPv6-Tunnel/CN=fe00');
14 exit 0 if ($cn eq '/C=DE/ST=Germany/O=hubertz-it-consulting_GmbH/OU=IPv6-Tunnel/CN=ff00');
15 exit 1;
```

Remark: one server-instance for each possible client connection



IPv6 transportation via IPv4

OpenVPN is our friend, `client-config`⁷ follows:

```
1 dev tun
2 tun-ipv6
3 remote 192.0.2.126 # Test-Net of RFC 3330 - Special-Use IPv4-Addresses
4 port 4712
5 proto tcp-client
6 script-security 2
7 up /etc/openvpn/client-up-script
8 down /etc/openvpn/client-down-script
9 ca /etc/openvpn/cacert.pem
10 cert /etc/openvpn/cert.pem
11 key /etc/openvpn/key.pem
12 tls-client
13 tls-verify /etc/openvpn/tls-verify
14 log-append /dev/shm/openvpn.log
15 persist-tun
16 ping 15
17 ping-restart 45
18 ping-timer-rem
19 persist-key
20 verb 3
```

⁷Linux and OpenBSD



IPv6 transportation via IPv4

OpenVPN is our friend, `client-up-script`⁸ follows:

```
1 #!/bin/bash
2
3 INTERFACE=$1; shift;
4 TUN_MTU=$1; shift;
5 UDP_MTU=$1; shift;
6 LOCAL_IP=$1; shift;
7 #REMOTE_IP=$1; shift;
8 #MODUS=$1; shift;
9
10 /usr/sbin/openvpn --mktun --dev ${INTERFACE}
11 /sbin/ip link set ${INTERFACE} up
12 /sbin/ip link set mtu ${TUN_MTU} dev ${INTERFACE}
13 /sbin/ip addr add fe80::fc00::2/64 dev ${INTERFACE}
14 /sbin/ip route add 2000::/3 via fe80::fc00::1 dev ${INTERFACE}
15 /sbin/ip addr add 2001:db8:f002:fc00:4711::1/128 dev eth2
16 /sbin/ip addr add 2001:db8:f002:fc00::1/64 dev eth2
17 /sbin/ip route add 2001:db8:f002:fc00::/56 via 2001:db8:f002:fc00::2
18 exit 0
```

Remark: Line 15 assures to have least one global IP independant from status of eth2, needed for keepalive

⁸Linux



IPv6 transportation via IPv4

OpenBSD and OpenVPN are our friends
configuration is similar except: **client-up-script**⁹

```
1 #!/bin/bash
2
3 INTERFACE=$1; shift;
4 TUN_MTU=$1; shift;
5 UDP_MTU=$1; shift;
6 LOCAL_IP=$1; shift;
7 #REMOTE_IP=$1; shift;
8 #MODUS=$1; shift;
9
10 /sbin/ifconfig ${INTERFACE} up
11 /sbin/ifconfig ${INTERFACE} mtu ${TUN_MTU}
12 /sbin/ifconfig ${INTERFACE} inet6 fa80:0:fc00::2/64
13 /sbin/route -n add -inet6 2001:: -prefixlen 3 -link tun0
14 exit 0
```

Remark: sometimes keepalive fails for unknown reason, => not yet ready for distribution

⁹OpenBSD



Quellen und Lesestoff

... only a few of more than 200 ...

- RFC 2460 Internet Protocol, Version 6 (IPv6) Specification
- RFC 2461 Neighbor Discovery for IP Version 6 (IPv6)
- RFC 2462 IPv6 Stateless Address Autoconfiguration
- RFC 2463 Internet Control Message Protocol for the Internet Protocol Version 6 (IPv6) Specification
- RFC 2464 Transmission of IPv6 Packets over Ethernet Networks
- RFC 3315 Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- RFC 3756 IPv6 Neighbor Discovery (ND) Trust Models and Threats
- RFC 3775 Mobility Support in IPv6
- RFC 3971 SEcure Neighbor Discovery (SEND)
- RFC 3972 Cryptographically Generated Addresses (CGA)
- RFC 4429 Optimistic Duplicate Address Detection (DAD) for IPv6
- RFC 4443 Internet Control Message Protocol for the Internet Protocol Version 6 (IPv6) Specification
- RFC 4861 Neighbor Discovery for IPv6
- RFC 4890 Recommendations for Filtering ICMPv6 Messages in Firewalls
- RFC 5095 Deprecation of RH0

Linux:

<http://www.bieringer.de/linux/IPv6/IPv6-HOWTO/IPv6-HOWTO.html>
OpenVPN-tunnelbroker: <http://blog.ghitr.com/index.php/archives/673>
<http://www.6net.org/publications/presentations/strauf-openvpn.pdf>

Books:

IPv6 in Practice, Benedikt Stokebrand, Springer, ISBN 978-3-540-24524-7
IPv6, Sylvia Hagen, Sunny Edition, 2. Auflage, ISBN 978-3-9522842-2-2
Deploying IPv6 Networks, Ciprian Popoviciu et.al., Cisco Press, ISBN 1587052105

Tests:

<http://freeworld.thc.org/thc-ipv6/>
<http://lg.he.net/>

Security:

http://www.wecon.net/files/48/GUUG-RT_WEST2010-SvI.pdf
<http://seanconvery.com/ipv6.html>



Ich bedanke mich für Ihre Aufmerksamkeit

hubertz-it-consulting GmbH jederzeit zu Ihren Diensten

Ihre Sicherheit ist uns wichtig!

Frohes Schaffen

Johannes Hubertz

it-consulting _at_ hubertz dot de

H-alpha \in { kompetenzspektrum.de }



powered by **L^AT_EX 2_ε**
and PSTricks

