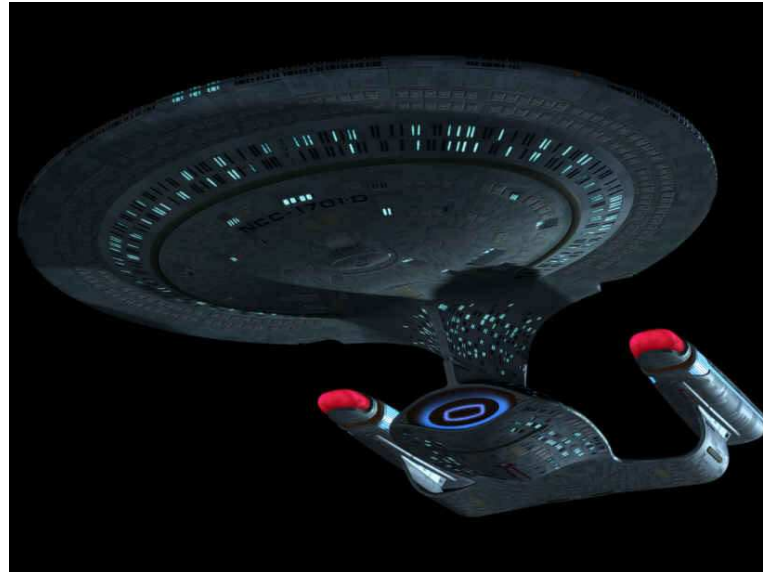


IPv6 / IPng



IPv6 / IPng



The Next Generation Internet Protocol
oder
„Wie man die Turtle zum Tanzen bringt“

Agenda

- Warum IPv6?
 - Die meistgenannten Argumente (Für und Wieder)
 - Keine „Killerapplikation“ (Eher: Was geht, und was nicht)
- IPv6 Konzepte
 - Adresskonzept
 - Autokonfiguration
 - DNS
- Implementierungen / Applikationen
- Migrationsszenarien
 - Tunnel: 6to4
 - NATPT, IPv4/IPv6 Proxy
- Zusammenfassung / Demo ?

IP Version 6

- 128 Bit Adressraum erlaubt (wieder) Ende-zu-Ende-Adressierung
Kein NAT, PAT oder ähnlicher M*st
- Vereinfachter Headeraufbau sorgt für performanteres Forwarding
Konstante (IPv6-)Headerlänge (Aber heute wird sowieso nur noch geswitcht)
- Integrierter Security Header (IPSEC: AH+ESP) (mandatory)
Key Management ist bei IPsec immer noch das größte Problem
- Mobile IPv6
 - Unterbrechungsfreie(!) mobile Kommunikation
 - Bewegte Hosts / Netze (IPv6 Car)
 - Keine Foreign-Agents notwendig
 - Optimiertes Routing (Kein „Triangle“-Routing)
- Stateless/Statefull Autoconfig
Einfaches Renumbering möglich, auch ohne DHCP

IP Version 6 (2)

- Kein NAT: Firewall notwendig (draft-vandevelde-v6ops-nap-00.txt)
(Wobei NAT sowieso keine Firewall ersetzt!)
- Kein NAT: Mehr P2P-Applikationen
 - Erlaubt neue Applikationen (VoIP, Network Games)
 - Sicherheitskonzept ändert sich vom Edge zum Host
 - „Firesuite“ anstelle einer Firewall
 - Windows XP SP2: IPv6 + Host Firewall
- Zusätzliche Komplexität durch mehrere Address-Families (inet/inet6)
 - Transition Problem
 - Programmierung der Applikation aufwändiger/fehleranfälliger
 - Unterschiedliche Unterstützung durch das OS
 - Noch keine einheitliche API etabliert (RFC3493, RFC3542)

IPv6 Adressaufbau (RFC3513)

- 128Bit, dargestellt in Hexnotation, je zwei Byte durch Doppelpunkt getrennt, führende Nullen können weggelassen werden

```
2001:0db8:0001:0000:0000:0000:00C0:007A  
2001:db8:1:0:0:0:C0:7A
```

- Eine Folge von 0 Bits kann (einmalig) durch :: verkürzt werden

```
2001:db8:1::C0:7A  
FF01::101           Multicast  
::1                Loopback  
::1.3.68.3         IPv4 representation
```

- Netzmasken werden in Prefixschreibweise dargestellt

```
fe80::260:94ff:feb9:d826/10  
2001:db8:df5:1::/64  
2001:db8:df5:1:260:94ff:feb9:d826/128
```

- Es gibt Unicast, Anycast und Multicast Adressen
Keine Broadcastadressen!

IPv6 Adressaufbau (Unicast)

- Typische Aufteilung (globale Unicast-Adressen) (RFC2928):

128 Bit				
Netzpart (64Bit)			Hostpart (64Bit)	
glob. Routingprefix (/48)			Kundenadressbereich (80Bit)	
001+ TLA	SubTLA(13) RIR, ISP, Customer	NLA(19)	SLA Subnet	Interface/Host ID
2001	0db8	1234	0001	0000:0000:0000:000c

- Jeder bekommt ein /48er IPv6-Netzwerkprefix
Bis zu 65535 Subnetze möglich
- Jedes Subnetz kann 2^{64} Hosts aufnehmen (18.446.744.073.709.551.616)

Hostadressvergabe:

- Statisch (z.B. 2001:db8:100::1)
- Automatisch aus der modifizierten MAC-Adresse (RFC3515)
(z.B. 2001:db8:100::260:94ff:feb9:d826)
- Cryptographisch generiert (Draft CGA)
- Zufällig (Draft Privacy Extensions)

IPv6 Unicast Adressen

- Link Local $\text{FE80}::/10$ (Hostpart 64 Bit)
- Site Local Adressen $\text{FEC0}::/10$
Deprecated (draft-ietf-ipv6-depreciat-site-local-04.txt)
- Private Adressbereiche $\text{FC00}::/7$
 - Centrally Assigned Unique Local Addresses $\text{FC00}::/8$
(draft-ietf-ipv6-ula-central-00.txt)
 - Unique Local IPv6 Unicast Addresses $\text{FD00}::/8$
(draft-ietf-ipv6-unique-local-addr-05.txt)
- Global Unicast Adressbereich (RFC2928) $2001::/16$
Vergabe von $/32$ an LIR (Local Internet Registrar)
- 6to4 global unicast Adressbereich $2002::/16$ (RFC 3056)
- 6BONE global unicast Adressbereich $3FFE::/16$ (RFC2471)
Adressvergabe wird eingestellt (draft-fink-6bone-phaseout-05.txt)

IPv6 Multicast Adressen

- Alle Adressen mit Prefix `FF00::/8` sind Multicastadressen

128 Bit			
8 Bit	4 Bit	4Bit	112 Bit
11111111	Flags	Scope	Gruppen-Adresse
FF	0	5	::101

- Flags bestimmen den Typ der Muticastadresse

0 Permanente Multicast Adresse z.B.

<code>FF02::1</code>	all nodes	<code>FF05::1:5</code>	all site DHCP-Servers
<code>FF02::2</code>	all routers	<code>FF0E::101</code>	all global NTP-Servers

3 Unicast-based Adressen (RFC3306)

7 Rendezvous-Point embedded Adressen

- Scope bestimmt die Reichweite („Sichtbarkeit“)

Wert	Scope	Wert	Scope
1	interface-local	5	site-local
2	link-local	8	organisation-local
4	admin-local	E	global

IPv4 versus IPv6

- IPv6 Adresse: Hexnotation durch Doppelpunkt getrennt (Manchmal auch in eckigen Klammern)

```
http://1.112.15.4:8080/index.html
```

```
http://[2001:db8:5ff::eb]:8080/index.html
```

```
scp bsp.file hoz@[2001:db8:5ff::5cb]:/tmp
```

- Subnetzmaske ist immer /64!
Aber: Keine Regel ohne Ausnahme!
- HostID ist unabhängig vom Subnetz!
- Ein IPv6 Host hat meist mehrere IP-Adressen pro Interface
z.B. 2001 u. 2002 oder Home-Address und „Care-of-Address“ bei MIPv6
- Anstelle von ARP erweitertes ICMP (Neighbor Discovery)
- Netzprefix und default Gateway(s) werden vom Router propagiert
Kein DHCP, bzw. HSRP oder VRRP notwendig

Address Allocation Policy

- IANA bzw. IAB an RIR (z.B. RIPE NCC) (RFC 2928)
 - Unter TLA 2001::/16 z. Zeit mehrere /29
- RIR an LIR (z.B. Arcor AG) (ripe-267)
 - Minimum /32 d.h. $2^{(48-32)} = 65536$ Kundennetze
- LIR an Site (RFC 3177)
 - Home Networks, small and large Enterprise /48
Normalfall(!), ausgenommen wirklich große Sites (s.u.)
 - Very large Subscribers /47 oder mehrere /48
 - Mobile Networks mit einem Subnetz /64
(z.B. Mobile Phone mit Bluetooth-Adapter und mehreren Geräten)
 - Einzelner PC z.B. in einem Hotel /128
- Keine PI-Adressen!

Stateless Autoconfiguration (RFC2462)

- Verfahren zur automatischen Vergabe einer (global routingfähigen) IP-Adresse ohne einen externen Dienst wie z.B. DHCP
- Prinzipieller Ablauf (vereinfacht):
 - a. Host bildet aus seiner MAC-Adresse den Interface-Teil einer IPv6 Adresse (64Bit)
Oder „Privacy Extensions for Stateless Autoconfiguration“ RFC3041
 - b. Daraus wird mit dem Prefix $FE80::/10$ eine Link-Local-Adresse
 - c. Über den *Duplicate Address Detection* (DAD) Mechanismus wird für eine eindeutige Link-Local-Adresse gesorgt
 - d. Empfängt der Host Router Advertisement Messages, wird mit Hilfe des ausgesendeten Prefixes eine (oder mehrere) global routingfähige Adresse gebildet
 - e. Die Router Advertisement Messages werden auch für die Konfiguration einer Default-Route verwendet

Statefull Autoconfiguration (DHCP)

- Statefull Autoconfiguration basiert auf DHCP für IPv6 (RFC3315)
DHCPv4 und DHCPv6 unterscheiden sich grundlegend
- Ein Bit in der Router Advertisement Message kennzeichnet ob statefull autoconfig genutzt werden soll
- Bietet die Möglichkeit Service-Adressen zu übermitteln
z.B. DNS-Server
- Erlaubt die automatische Registrierung der Hosts im Reverse-DNS
- Genauere Info:
„DHCPv6“, Nov. 2003, 39. DFN Betriebstagung C. Strauf, Uni-Münster
<http://www.join.uni-muenster.de/Dokumente/Folien/strauf/39.DFN-BT/DHCPv6/DHCPv6.pdf>

DNS

- Die Namensauflösung von IPv6-Adressen erfolgt über AAAA-Records
A6-Records sind als experimental deklariert worden (RFC3596)

```
$ORIGIN example.de.  
host      IN      A        1.2.3.4  
          IN      AAAA     2001:db8:400:1:210:5aff:feac:8b1b  
          IN      AAAA     2001:db8:304:3f56:210:5aff:feac:8b1b
```

- Reverse Zone liegt unter `ip6.arpa` und **nicht** unter `ip6.int` (RFC3596, RFC3152, BCP49)

```
$ORIGIN 1.0.0.0.0.4.0.8.b.d.0.1.0.0.2.ip6.arpa.  
b.1.b.8.c.a.e.f.f.f.a.5.0.1.2.0      PTR      host.example.de.
```

- Aufteilung der Netze an /64 Grenze vereinfacht die Reverse DNS-Konfiguration (Immer 16 Nibbles für's Netz und 16 für den Host)
- Nur globale, permanente unicast Adressen im DNS aufnehmen
Keine Link-Local-, keine Private- oder zufällig gewählte Adressen

DNS (2)

- Sind für einen Host sowohl IPv4 als auch IPv6 Adressen im DNS hinterlegt **muß** der Host auch von **allen** Hosts die diese Information bekommen über IPv4 **und** IPv6 erreichbar sein!
- DNAME-Records sind **nicht** als „experimental“ eingestuft
Aus Portabilitätsgründen empfiehlt sich ein
`x.y.ip6.int DNAME x.y.ip6.arpa.`
insofern `ip6.int` überhaupt delegiert wird
- DNS-Registrierung der Hosts aufwändiger als bei IPv4 aber notwendig!
(Inklusive Reverse-Auflösung!)

Beispiel:

```
$ dig +noall +answ +cmd aaaa qeylIS.klingons.de
; <<>> DiG 9.3.0 <<>> +noall +answ +cmd aaaa qeylIS.klingons.de
;; global options:  printcmd
qeylIS.klingons.de.      86361 IN      AAAA  2001:1b20:1:4::c
```

```
$ dig +noall +answ -x 2001:1b20:1:4::c
c.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.0.0.0.1.0.0.0.0.2.b.1.1.0.0.2.ip6.arpa. \
                                                                85080 IN PTR qeylIS.klingons.de.
```

IPv6 Implementierungen (Auswahl)

- Apple Mac OS 10.2
- Sun Solaris 2.7/8/9; 6to4 ab Solaris9 12/03
- Linux (ab Kernel 2.2), RedHat 7.2, SuSE 8.1
- NetBSD (ab 1.5) / FreeBSD (ab 4.0) / OpenBSD (ab 2.8)
- Microsoft Windows Server 2003 und Windows XP ab SP1
„Microsoft Windows XP SP1 includes a production-quality version of the IPv6 protocol.“
Einschalten über

```
c:> ipv6 install
```

- Cisco IPv6
6to4: 12.3(2)T und 12.0S; 6PE: 12.2S bzw. 12.0(26)S
- Juniper (JUNOS 5.2)
- Ericsson
- Nokia

Weitere Hinweise über [IPv6 Forum](http://www.ipv6forum.com)¹ unter „Implementations“

1. <http://www.ipv6forum.com>

Applikationen

http://www.deepspace6.net/docs/ipv6_status_page_apps.html

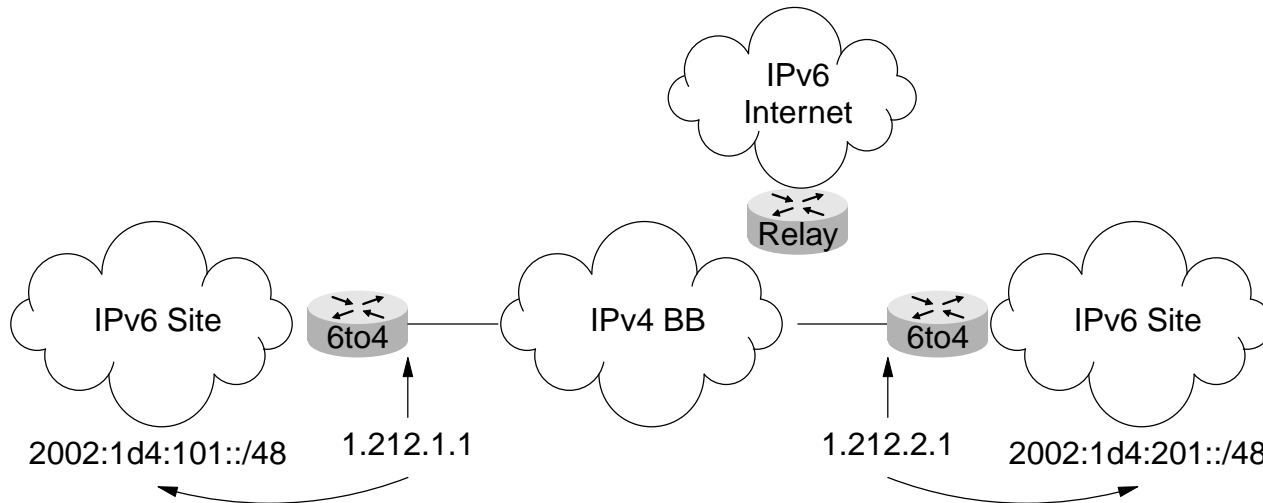
<i>DNS</i>	BIND9, djbdns+patches, nsd	totd
<i>Web</i>	Apache (2.0), thttpd	Mozilla, Konqueror, lynx, wget, curl
<i>SMTP</i>	sendmail	fetchmail
<i>IMAP, POP3</i>	courier, cyrus	mutt, fetchmail
<i>LDAP</i>	openldap	
<i>Filetransfer</i>	proftpd, tftpd rsync, scp, sftp	ftp, tftp
<i>Sicherheit</i>	SSH, OpenSSH Kerberos (Heimdal) tcp_wrapper, Socks5 (NEC)	Cyrus SASL
<i>Routing</i>	GateD, Zebra, Bird	MRTd
<i>Information</i>		whois, finger
<i>Systemutilities</i>	(x)ntpd xinetd, syslog-ng dhcpv6, radvd	ntpdate ping6, traceroute6, nc6 nmap, nessus tcpdump, ethereal
<i>Multimedia</i>	gnomemeeting(VoIP) vls (VideoLan)	xmms (mp3),Mplayer (Movie) vlc

Tunnellösungen

- Manuelle Tunnel
Einfache, stabile und sichere Verbindung zweier Endpunkte.
 - IPv6 in IPv4
 - IPv6 over IPv4 GRE (Für IS-IS Routing im Tunnel notwendig)
- Halbautomatische Tunnel (Tunnelbroker <http://www.sixxs.net>)
Software für skriptbasierte Konfiguration des Routers.
- Automatische Tunnel
 - IPv4 compatible (deprecated)
Schlechte Skalierung, jeder Host benötigt eindeutige IPv4-Adresse.
 - 6to4 (RFC3056)
Einfach zu managen, skalierbar, nur eine öffentliche IPv4-Adresse notwendig
- Enterprise/Campus Tunnellösungen
 - 6over4 (benötigt IPv4 Multicast)
 - ISATAP (Intra-Site Automatic Tunnel Access Protocol)
(draft-ietf-ngtrans-isatab-12.txt)

6to4 (RFC 3056)

- Pro Site eine öffentliche IPv4-Adresse notwendig
- Network-Prefix für alle IPv6Hosts dieser Site: $2002:IPv4-Addr::/48$
- 6to4 Router (CPE) tunnelt alle $2002::/16$ Netze über IPv4 zu den entsprechenden IPv4-6to4-Routern.
- Native IPv6 Konnektivität über 6to4 Relay-Router.



6to4 (2)

- IPv4 Adresse sollte statisch sein
Sonst kann man keine Server betreiben :-)

```
interface Tunnel2002
  description 6to4 tunnel (Prefix 2002:1.212.2.1::1)
  ipv6 address 2002:1d4:201::1/128
  tunnel source Ethernet0
  tunnel mode ipv6ip 6to4
```

- Transparenter Datenverkehr für IP-Protocol 41 (IPv6) notwendig
- Default IPv6-Route zeigt auf 6to4-Relay

```
ipv6 route 2002::/16 tunnel2002
ipv6 route ::/0 2002:c048:2101::1
```

- Relay-Router mit Anycast-Adresse (RFC3068) 192.88.99.0/24

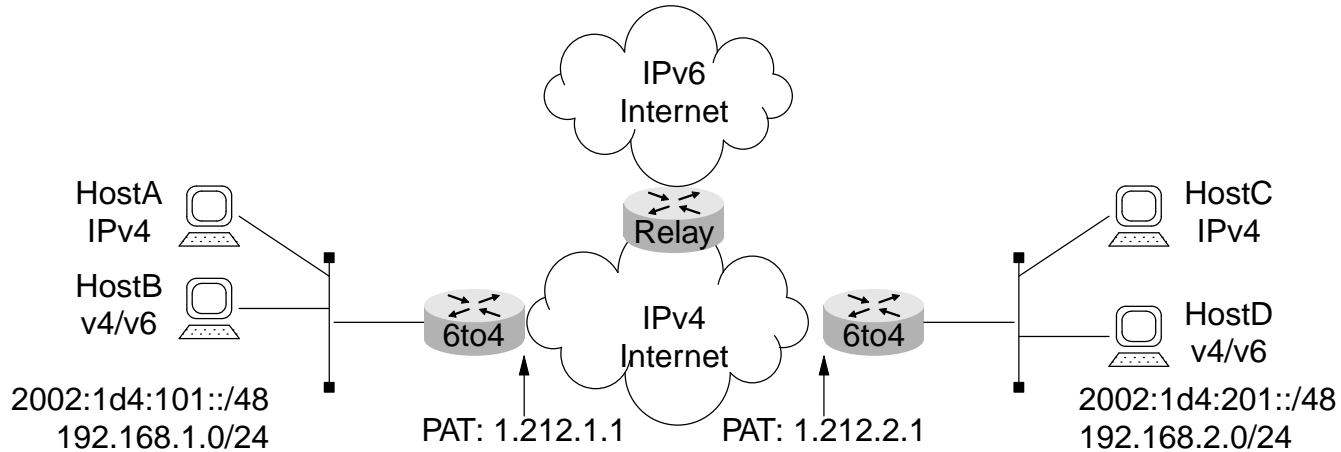
```
ipv6 route ::/0 2002:c058:6301::1
```

- Relay-Router propagiert 2002::/16-Route im IPv6-Internet
Kein längerer Routing-Prefix!

- Asymmetrisches Routing
Rücktraffic kommt über anderen Relay-Router

6to4 mit IPv4-PAT

DNS left		DNS global		DNS right	
HostA	A	HostB	AAAA	HostB	AAAA
HostB	A+AAAA	HostD	AAAA	HostC	A
HostD	AAAA			HostD	A+AAAA



Kommunikationsbeziehungen					
IPv4			IPv6		
HostA	↔	HostB	HostB	↔	HostD
HostC	↔	HostD			
A,B,C,D	⇒	IPv4-Internet	HostB,HostD	↔	IPv6-Internet

6to4 — Zusammenfassung

- Jeder kann IPv6 (Standort übergreifend) nutzen
- Keine externen Abhängigkeiten („*Mein Provider kann kein IPv6*“ o.ä.)
- Änderungen der CPE-Konfig marginal
Dual-Stack fähiges IOS, ein Tunneldevice, zwei zusätzliche Routen, radvd
- Keine Änderungen am Backbone; Unterstützung bel. Accesstechniken
DSL, Dial, WLAN
- 6to4-Relay Router für Verbindung zum IPv6-Internet notwendig
Typischerweise durch ISP bereitgestellt
- IPv6 fähige CS-Dienste
DNS, MTA, HTTP-Proxy, SIP
- Langfristig „echte“ IPv6-Adressen und IPv6-Konnektivität gefordert

Protocol Translation

- IPv6 zu IPv4 Kommunikation über Protokoll Umsetzung Erlaubt native IPv6 zu native IPv4 Kommunikation.
- Notwendig wenn IPv6-only Netze angebunden werden.
- Diverse Mechanismen
 - SIIT (Stateless IP/ICMP Translation RFC2765)
 - NAT-PT (Umsetzung auf Network Ebene RFC2766)
Router Implementierung + DNS Application Level Gateway
 - TCP/UDP Relay (Protokoll Umsetzung auf Transport Ebene)
faithd, totd, pTRTd (RFC3142)
 - SOCKS64 (APL-Gateway + Software auf dem Client RFC3089)
Für alle Dienste nutzbar(?), gleichzeitig als Firewall einsetzbar!
 - Proxy (Dual-Stack Applikation Level Gateway)
Für jeden Dienst separate Software bzw. Server (squid, sendmail).
- IPv4 zu IPv6 Protokollumsetzung
 - Nur Proxylösungen sinnvoll (z.B. ffmproxy)

Zusammenfassung

- Einfacher Einstieg in IPv6 über 6to4 oder manuelle Tunnel möglich.
 - Kein IPv6-Netz notwendig
 - Microsoft forciert 6to4 und Teredo (6to4 hinter PAT)
- Erste Erfahrungen mit IPv6 sammeln
Einfach einschalten (c:> ipv6 install)
- Die wichtigsten Dienste sind IPv6 enabled
 - HTTP Client/Server
 - MTA
 - VoIP (allerdings wenige Clients)
 - andere?IPv6 einschalten!
- Transparente Kommunikation zwischen NAT-Standorten möglich

References

Cisco

„IPv6 Deployment Strategies“, Cisco, Mai 2003
(<http://www.cisco.com/...../ipv6dswp.htm>)

Sylvia Hagen

„IPv6 Essentials“, O'Reilly & Associates, Juli 2002

Paul Albitz, Cricket Liu

„DNS and BIND“, O'Reilly & Associates, 1998

Links

- IPv6 Test Backbone <http://6bone.net>
- IPv6 Infopage <http://www.ipv6.org>
- Weltweites IPv6 Forum <http://www.ipv6forum.com>
- IETF Working Group [http://www.ietf.org/html.charters/ipv6\[ops\]-charter.html](http://www.ietf.org/html.charters/ipv6[ops]-charter.html)
- IPv6 Programm der EU (Jan 2002 bis Dez 2004) <http://www.6net.org>
- Cisco <http://www.cisco.com/go/ipv6>
- Microsoft <http://www.microsoft.com/ipv6>

RFC

Diverse

Fragen ?

Fragen ?

<http://www.hznet.de/ipv6/ipv6-einf.pdf>

Fragen ?

<http://www.hznet.de/ipv6/ipv6-einf.pdf>

Herzlichen Dank für Ihre Aufmerksamkeit

Cisco IOS – 6to4 Konfiguration

- IPv6 Routing einschalten und ext. IPv4 Interface konfigurieren

```
ipv6 unicast-routing  
  
interface FastEthernet0  
  description external ipv4 interface  
  ip address 145.253.100.38 255.255.255.224
```

- 6to4 Tunnelinterface anlegen:

```
interface Tunnel2002  
  description 6to4 tunnel  
  no ip address  
  ipv6 address 2002:91FD:6426::1/128 # 2002:145.253.100.63::1  
  tunnel source FastEthernet0  
  tunnel mode ipv6ip 6to4
```

- „Lokales“ Interface für IPv4 und IPv6 konfigurieren:

```
interface Ethernet0  
  ip address 192.168.20.14 255.255.255.0  
  ipv6 address 2002:91FD:6426:1::1/64 Host 1 im Subnetz 1
```

- Statische Routen:

```
ip route 0.0.0.0 0.0.0.0 145.253.100.33  
ipv6 route 2002::/16 Tunnel2002  
ipv6 route ::/0 2002:C058:6301::1 # 2002:192.88.99.1::1
```

Linux IPv6 – 6to4 Konfiguration

- Ab Kernel 2.2 mit ip-route2 utilities

```
#!/usr/local/bin/bash
## use bash instead of bourne shell !
PATH=/bin:/usr/bin:/sbin:/usr/sbin

ip="1.2.3.4"      # replace with official ipv4 address

## calculate ipv6 prefix (2002:ip:v4)
ipbytes=`echo $ip | sed "s/\./ /g"`
ip6prefix=`printf 2002:%x%x:%x%x $ipbytes`

## create 6to4 tunnel device
ip tunnel add tun6to4 mode sit ttl 64 remote any local $ip
ip link set dev tun6to4 up
# the next command adds also a 2002:/16 route to tun6to4
ip -6 addr add ${ip6prefix}::1/16 dev tun6to4

## turn ip forwarding on
sysctl -w net.ipv6.conf.all.forwarding=1 > /dev/null

# set "default" ipv6 route (all global prefixes) to
# 6to4 relay router anycast addr (RFC3056)
ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4 metric 1
```

Linux IPv6 Test (1)

dig

```
$ dig +noall +cmd +answ any konstriktor.siski.de
; <<>> DiG 9.2.3rc2 <<>> +noall +cmd +answ any konstriktor.siski.de
;; global options:  printcmd
konstriktor.siski.de. 594333      IN      A       62.146.142.34
konstriktor.siski.de. 594333      IN      A6      48 ::1:0:0:0:1 siski.de.
konstriktor.siski.de. 594333      IN      AAAA    2002:3e92:8e22:1::1
```

IPv4 Ping

```
$ ping -n -c 2 konstriktor.siski.de
PING konstriktor.siski.de (62.146.142.34) 56(84) bytes of data.
64 bytes from 62.146.142.34: icmp_seq=1 ttl=53 time=46.1 ms
64 bytes from 62.146.142.34: icmp_seq=2 ttl=53 time=42.7 ms
--- konstriktor.siski.de ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 42.765/44.474/46.184/1.722 ms
```

```
$ tcpdump -n icmp
13:40:58.593458 145.253.100.37 > 62.146.142.34: icmp: echo request (DF)
13:40:58.640649 62.146.142.34 > 145.253.100.37: icmp: echo reply
13:40:59.604519 145.253.100.37 > 62.146.142.34: icmp: echo request (DF)
13:40:59.646335 62.146.142.34 > 145.253.100.37: icmp: echo reply
```

Linux IPv6 Test (2)

IPv6 Ping zu 6to4 host

```
$ ping6 -n -c 2 konstriktor.siski.de
PING konstriktor.siski.de(2002:3e92:8e22:1::1) 56 data bytes
64 bytes from 2002:3e92:8e22:1::1: icmp_seq=1 ttl=64 time=44.7 ms
64 bytes from 2002:3e92:8e22:1::1: icmp_seq=2 ttl=64 time=43.3 ms
--- konstriktor.siski.de ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 43.342/44.041/44.741/0.730 ms

$ tcpdump -n proto 41 or proto 58
... 145.253.100.37 > 62.146.142.34: 2002:91fd:6425::1 > 2002:3e92:8e22:1::1: icmp6: echo request (DF)
... 62.146.142.34 > 145.253.100.37: 2002:3e92:8e22:1::1 > 2002:91fd:6425::1: icmp6: echo reply (DF)
... 145.253.100.37 > 62.146.142.34: 2002:91fd:6425::1 > 2002:3e92:8e22:1::1: icmp6: echo request (DF)
... 62.146.142.34 > 145.253.100.37: 2002:3e92:8e22:1::1 > 2002:91fd:6425::1: icmp6: echo reply (DF)
```

IPv6 Ping zum 6Bone (via 6to4-Relay Router)

```
$ ping6 -n -c 2 www.6bone.net
PING www.6bone.net(3ffe:b00:c18:1::10) 56 data bytes
64 bytes from 3ffe:b00:c18:1::10: icmp_seq=1 ttl=61 time=383 ms
64 bytes from 3ffe:b00:c18:1::10: icmp_seq=2 ttl=61 time=385 ms
--- www.6bone.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 383.223/384.525/385.828/1.442 ms

$ tcpdump -n proto ipv6 or proto ipv6-icmp
... 145.253.100.37 > 192.88.99.1: 2002:91fd:6425::1 > 3ffe:b00:c18:1::10: icmp6: echo request (DF)
... 202.39.142.146 > 145.253.100.37: 3ffe:b00:c18:1::10 > 2002:91fd:6425::1: icmp6: echo reply
... 145.253.100.37 > 192.88.99.1: 2002:91fd:6425::1 > 3ffe:b00:c18:1::10: icmp6: echo request (DF)
... 202.39.142.146 > 145.253.100.37: 3ffe:b00:c18:1::10 > 2002:91fd:6425::1: icmp6: echo reply
```


Microsoft IPv6 und 6to4 Konfiguration

- Windows 2003 und XP (ab SP1) unterscheiden sich
Alle folgenden Angaben beziehen sich auf XP

- IPv6 aktivieren (falls noch nicht geschehen)

```
c:> ipv6 install
```

- Kontrollieren:

```
c:> netsh int ipv6 show interface # v6 Interface anzeigen  
c:> netsh int ipv6 show routes # v6 Routen anzeigen
```

- Evtl. 6to4 einrichten

```
c:> netsh int ipv6 6to4 # Einrichten  
c:> netsh int ipv6 6to4 set relay # Relay Router konfigurieren
```

- Testen:

```
c:> ping6 m6x.hznet.de  
c:> tracert6 m6x.hznet.de
```

Achtung: Webbrowser bevorzugt IPv4 vor IPv6 bei Dualstack Webserver
Zum Testen einen IPv6 only Webserver verwenden (z.B. <http://www.ipv6.uni-muenster.de>)

CONTENTS

.....	1
Agenda	2
IP Version 6	3
IP Version 6 (2)	4
IPv6 Adressaufbau (RFC3513)	5
IPv6 Adressaufbau (Unicast)	6
IPv6 Unicast Adressen	7
IPv6 Multicast Adressen	8
IPv4 versus IPv6	9
Address Allocation Policy	10
Stateless Autoconfiguration (RFC2462)	11
Statefull Autoconfiguration (DHCP)	12
DNS	13
DNS (2)	14
IPv6 Implementierungen (Auswahl)	15
Applikationen	16
Tunnellösungen	17
6to4 (RFC 3056)	18
6to4 (2)	19
6to4 mit IPv4-PAT	20
6to4 — Zusammenfassung	21
Protocol Translation	22
Zusammenfassung	23
References	24
.....	25
Cisco IOS – 6to4 Konfiguration	26
Linux IPv6 – 6to4 Konfiguration	27
Linux IPv6 Test (1)	28
Linux IPv6 Test (2)	29
Microsoft IPv6 und 6to4 Konfiguration	30