

## Wat basiskennis...

<i>TCP/IP model</i>	<i>OSI model</i>
Applicatie	Applicatie Presentatie Sessie
Transport	Transport
Internet	Netwerk
Fysiek	Data Link Fysiek

- ▶ MAC (bv: 90:fb:a6:ae:b3:5a)
- ▶ IPv4 (bv: 127.0.0.1)
- ▶ (R)ARP
- ▶ TCP, UDP, ...
- ▶ HTTP, FTP, ...

Notities

---

---

---

---

---

---

---

---

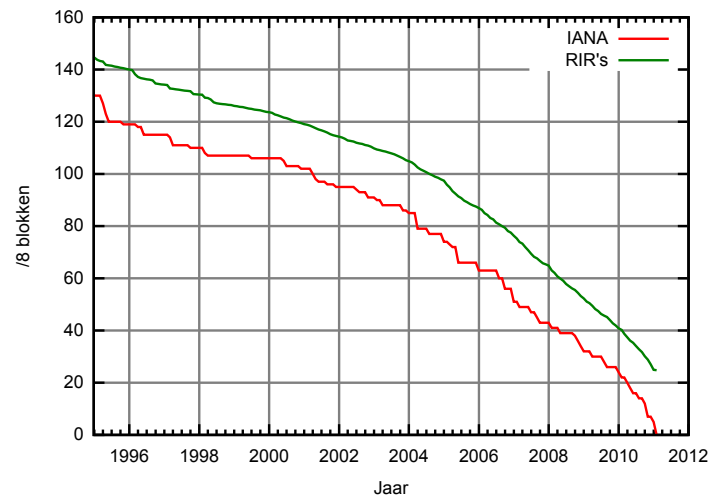
---

---

## IPv4, is het einde nabij ?

IANA → RIR's → LIR's → ISP's → Klanten

Vrije /8 blokken



Notities

---

---

---

---

---

---

---

---

---

---

# Oplossingen

- ▶ IPv6 ?
- ▶ NAT ?
- ▶ Opeisen ongebruikte IPv4 klasse A's ?
- ▶ Splitsen ?

Notities

---

---

---

---

---

---

---

---

---

---

# Schrijfwijze IPv6 adressen

- ▶ 128 bits = 68 jaar elke seconde iedereen een /64 subnet extra
- ▶ 0012:0000:0000:0034:0000:0000:0fed:cba0/128
- ▶ 0012:0000:0000:0034:0000:0000:0fed:cba0 =  
12:0:0:34:0:0:fed:cba0
- ▶ 12:0:0:34:0:0:fed:cba0 = 12::34:0:0:fed:cba0
- ▶ ::123.45.67.89 IPv4-compatibel, Pakketjes richting dit adres zijn IPv6 met IPv4 header en dus **niet** begrepen door systemen zonder IPv6 stack. (Deprecated sinds 02/2006 !)
- ▶ ::ffff:123.45.67.89 IPv4-mapped, Pakketjes richting dit adres worden gewoon als IPv4 pakket verstuurd. (Nutteloos en niet altijd ondersteund)
- ▶ IPv6-adres met TCP/UDP poort combinatie heeft het adres normaal tussen rechte haakjes. bv.: [::1]:123

Notities

---

---

---

---

---

---

---

---

---

---

# Unicast

48 bits	16 bits	64 bits
netwerkprefix		interface id
routingprefix	subnet id	

- ▶ Internet 2000::/3 (zoals 0.0.0.0/0)
- ▶ Localhost ::1 (zoals 127.0.0.0/8)
- ▶ Lokaal fc00::/7 (zoals 10.0.0.0/8, 172.16.0.0/12 en 192.168.0.0/16)
- ▶ Link-Local fe80::/10 eigenlijk fe80::/64 (zoals 169.254.0.0/16)
- ▶ Default route ::/0 (zoals 0.0.0.0/0)
- ▶ 6to4 tunnels 2002::/16
- ▶ ... (Documentatie, Torpedo tunnels, BMWG, ORCHID, ...)

Notities

---

---

---

---

---

---

---

---

---

---

# EUI-64 type adressen

- ▶ Stel: MAC-adres is 01:23:45:67:89:ab
- ▶ Splits MAC-adres in 2 en zet er ff:fe tussen, dit geeft als EUI-64 01:23:45:ff:fe:67:89:ab
- ▶ 7de bit geeft uniekheid op internet weer, zet deze dus op 1
- ▶ Het IPv6 adres zal dus eindigen op :323:45ff:fe67:89ab

Een IPv6 in EUI-64 formaat eindigt niet noodzakelijk op ff:feXX:XXXX

Notities

---

---

---

---

---

---

---

---

---

---

# Multicast

ff00::/8 (zoals 224.0.0.0/4)

8 bits	4 bits	4 bits	112 bits
prefix (ff)	vlaggen	bereik	groep id

## Vlaggen

	0	1
<i>gereserveerd</i>	moet	/
<i>Rendezvous</i>	Niet in adres	in groep id
<i>Prefix</i>	Niet in adres	in groep id
<i>Transient</i>	Bekend (Statisch)	Dynamisch Gealloceerd

## Bereik

1	2	4	5	8	e	rest
interface	link	admin	site	organisatie	internet	/

Notities

---

---

---

---

---

---

---

---

---

---

---

---

# Enkele groep id's

(Met als voorbeeld ff02 voor, om niet te ver te multicasten)

- ▶ ff02::1 - Alles
- ▶ ff02::2 - Alle routers
- ▶ ff02::1:3 - Alle DHCP servers
- ▶ ff02::101 - Alle NTP servers
- ▶ ff02::1:ffXX:XXXX - Sollicited-node multicast

Om bijvoorbeeld te weten welke IPv6-sprekende NTP servers direct aangesloten zijn op eth0 doe: ping6 -I eth0 ff02::101

Notities

---

---

---

---

---

---

---

---

---

---

---

---

## Verdeling adressen

- ▶ IANA → RIR's: /23 tot /12 (/8 bij IPv4)
- ▶ RIR's → LIR's: /32 (/20 tot /8 bij IPv4)
- ▶ LIR's → ISP's: /35 tot /32, max /29 (meestal /16 bij IPv4)
- ▶ ISP's → Klanten: /48 tot /64 (meestal /24 tot /32 bij IPv4)

Notities

---

---

---

---

---

---

---

---

---

---

## Neighborhood Discovery

- ▶ Vervangt oa. ARP
- ▶ Stel: We willen 2001::1 bereiken
- ▶ Stuur een ND pakketje dat vraagt waar 2001::1 is naar het Solicited-node multicast adres
- ▶ Dit adres is ff02::1:ffXX:XXXX, met als laatste 24 bits de laatste 24 bits van het te bereiken adres
- ▶ In ons geval dus ff02::1:ff00:1
- ▶ Iedereen op de link met 00:0001 op het einde van zijn adres ontvangt dit pakket, maar in het ND pakket zelf is 2001::1 vermeld, dus alleen hij antwoord

Notities

---

---

---

---

---

---

---

---

---

---

## Header

- ▶ Versie, 4 bits 0110
- ▶ DSCP en ECN, 6 bits en 2 bits
- ▶ Flow Label, 20 bits
- ▶ Payload lengte in bytes, 16 bits (0=jumbogram)
- ▶ Volgende header, 8 bits
- ▶ Hop Limit, 8 bits (IPv6 versie van TTL)
- ▶ Source, 128 bits
- ▶ Destination, 128 bits

Totaal pakket: 40 bytes tot 4 gigabyte  
(20 bytes tot 66 kilobytes bij IPv4)

Notities

---

---

---

---

---

---

---

---

---

---

## Fragmentatie

- ▶ Zoveel mogelijk overlaten aan bovenliggende lagen
- ▶ Anders de endpoints, routers fragmenteren niet (op inetlaag)

Notities

---

---

---

---

---

---

---

---

---

---



## Verbinden met internet

- ▶ Providers ?
- ▶ Zelf een 6to4 tunnel maken
- ▶ Tunnel brokers : [http://en.wikipedia.org/wiki/List\\_of\\_IPv6\\_tunnel\\_brokers](http://en.wikipedia.org/wiki/List_of_IPv6_tunnel_brokers)

Notities

---

---

---

---

---

---

---

---

---

---

## Een 6to4 tunnel maken op Linux

Stel: De kaart verbonden met internet is 12.34.56.78

1. Zoek het juiste IPv6 adres: 2002:c22:384e::1 (2002 = 6to4, c22:384e = het IPv4 adres, laatste 80 bits = eigen keuze)
2. Maak de tunnel: `ip tunnel add mijn6to4 mode sit remote any local 12.34.56.78 ttl 64`
3. Zet hem aan: `ip link set dev mijn6to4 up`
4. Geef hem het IPv6 adres:  
`ip -6 addr add 2002:c22:384e::1/16 dev mijn6to4`
5. Toon hem de weg naar 't net: `ip -6 route add 2000::/3 via 2002:c058:6301::1 dev mijn6to4 metric 1`  
Momenteel kan men de gateway op de meeste distros ook als IPv4-compatibel adres schrijven (::192.88.99.1)

De tunnel wissen ? `ip -6 route flush dev mijn6to4 && ip link set dev mijn6to4 down && ip tunnel del mijn6to4`

Notities

---

---

---

---

---

---

---

---

---

---



## Eigen Network

- ▶ Meerdere IP's per netwerkkaart mogelijk
- ▶ SLAAC
- ▶ DHCPv6
- ▶ Manueel

Notities

---

---

---

---

---

---

---

---

---

---

## StateLess Address AutoConfiguration

- ▶ Alles heeft bij het booten een link-local adres
- ▶ Stel MAC-adres is 01:23:45:67:89:ab, dus link-local is fe80::323:45ff:fe67:89ab
- ▶ DAD wordt gebruikt om te horen of niemand anders dit adres al gebruikt
- ▶ ND wordt gebruikt om netwerkconfiguratie op te vragen
- ▶ Router kan een DHCPv6 server aanwijzen maar zal meestal vertellen welke netwerk prefix te gebruiken (de interface id is die van de link-local)

Notities

---

---

---

---

---

---

---

---

---

---

## DHCPv6

- ▶ Clients luisteren op udp/546, Servers op udp/547
- ▶ 'Wild' of op basis van DUID
- ▶ Linux clients: Wide of Dobbler
- ▶ Linux servers: ISC of Dobbler
- ▶ Extra info

Momenteel 3 types DUID's, type 1 gebaseerd op link-laag-adres en de tijd. Zie sectie 9.3 en 9.4 van RFC 3315 voor de andere types.

- ▶ DUIDtype, 16 bits, 0001
- ▶ Hardwaretype, 16 bits, 0001 voor dingen met een MAC adres
- ▶ Link-laag-adres, lengte varieert (meestal MAC en dus 48 bits)

1 onveranderlijk DUID per systeem

## Manueel

Stel we willen eth0 het adres 2001::f geven in een /50 net

- ▶ `ifconfig eth0 inet6 add 2001::f/50`
- ▶ Resultaat: `inet6 addr: 2001::f/50 Scope:Global`
- ▶ Vergeet `ifconfig eth0 up` niet indien nodig
- ▶ `ifconfig eth0 inet6 del 2001::f/50` om het te wissen
- ▶ Kan ook met `ip`

Notities

---

---

---

---

---

---

---

---

---

---

Notities

---

---

---

---

---

---

---

---

---

---

## Routing

- ▶ Alle IPv6 routes tonen: `ip -6 route show`
- ▶ Alles voor internet via `2001::5` sturen:  
`ip -6 route add 2000::/3 via 2001::5`
- ▶ Routes wissen: `ip -6 route del 2000::/3 via 2001::5`
- ▶ `radvd`

Notities

---

---

---

---

---

---

---

---

---

---

---

## De Linuxkernel

- ▶ Instellingen in `/proc/sys/net/ipv6/conf/DEVICE/INSTELLING` (ook bereikbaar met `sysctl`). `DEVICE` kan ook `all` of `default` zijn.
- ▶ Belangrijkste `INSTELLING` is `forwarding`, deze geldt alleen voor `all`, gebruik `netfilter` voor details

Notities

---

---

---

---

---

---

---

---

---

---

---

## Mobile IPv6

- ▶ Als thuisadres een uniek maar GEEN link-local
- ▶ Care-of adres op andere locaties (mobiel systeem meld dit thuis)
- ▶ Mobiel systeem lijkt op 2 twee netwerken tegelijk te zitten
- ▶ Vreemde verbindingen eerst via thuis en daarna ook direct mogelijk.

Notities

---

---

---

---

---

---

---

---

---

---

## Veiligheid

- ▶ Firewall: netfilter6 geregeld met iptables
- ▶ Encryptie en authenticatie:
  - ▶ Altijd ondersteund, soms gebruikt
  - ▶ AES-128 of SHA1
  - ▶ Transport-modus: payload
  - ▶ Tunnel-modus: alles en nieuwe header

Notities

---

---

---

---

---

---

---

---

---

---

## Testen

IPv4 via directe verbinding met Telenet (Antwerpen) en IPv6 via tunnel van SixXs met als endpoint Easynet (Brussel) op dezelfde IPv4 verbinding

- ▶ `ping6 -c 50 -i 5 ipv6.google.com` : 2% verlies, 245s totaal, 48ms minimum, 50ms gemiddeld, 54ms maximum
- ▶ `ping -c 50 -i 5 google.com` : 10% verlies, 245s totaal, 14ms minimum, 16ms gemiddeld, 18ms maximum
- ▶ 50x Googlelogo (28KB) downloaden: 14 seconden over IPv6 en 8 seconden over IPv4
- ▶ 10x FreeBSD bootcd (51MB): 226 seconden via IPv6 en 215 seconden via IPv4

Notities

---

---

---

---

---

---

---

---

---

---

## Programmeren

Meeste dingen worden automatisch aangepast zolang dit maar gedaan wordt:

- ▶ `PF_INET` → `PF_INET6`
- ▶ `AF_INET` → `AF_INET6`
- ▶ `struct sockaddr_in` → `struct sockaddr_in6`
- ▶ mem-allocaties aanpassen als er te weinig `sizeof()-s` gebruikt worden.
- ▶ Adressen aanpassen

Notities

---

---

---

---

---

---

---

---

---

---

# Extra info

Zowat alles wat ik weet over IPv6 komt uit:

- ▶ RFC's
- ▶ Linux IPv6 HOWTO
- ▶ <http://en.wikipedia.org/wiki/Category:IPv6>
- ▶ O'Reilly boeken: IPv6 Essentials (2de editie), IPv6 Network Administration
- ▶ The Second Internet
- ▶ IRC kanaal #ipv6 op Freenode

Notities

---

---

---

---

---

---

---

---

---

---

Notities

---

---

---

---

---

---

---

---

---

---