

# IPv6



*Markus Schade*



**WE LOVE BITS. DO YOU?**

## Agenda

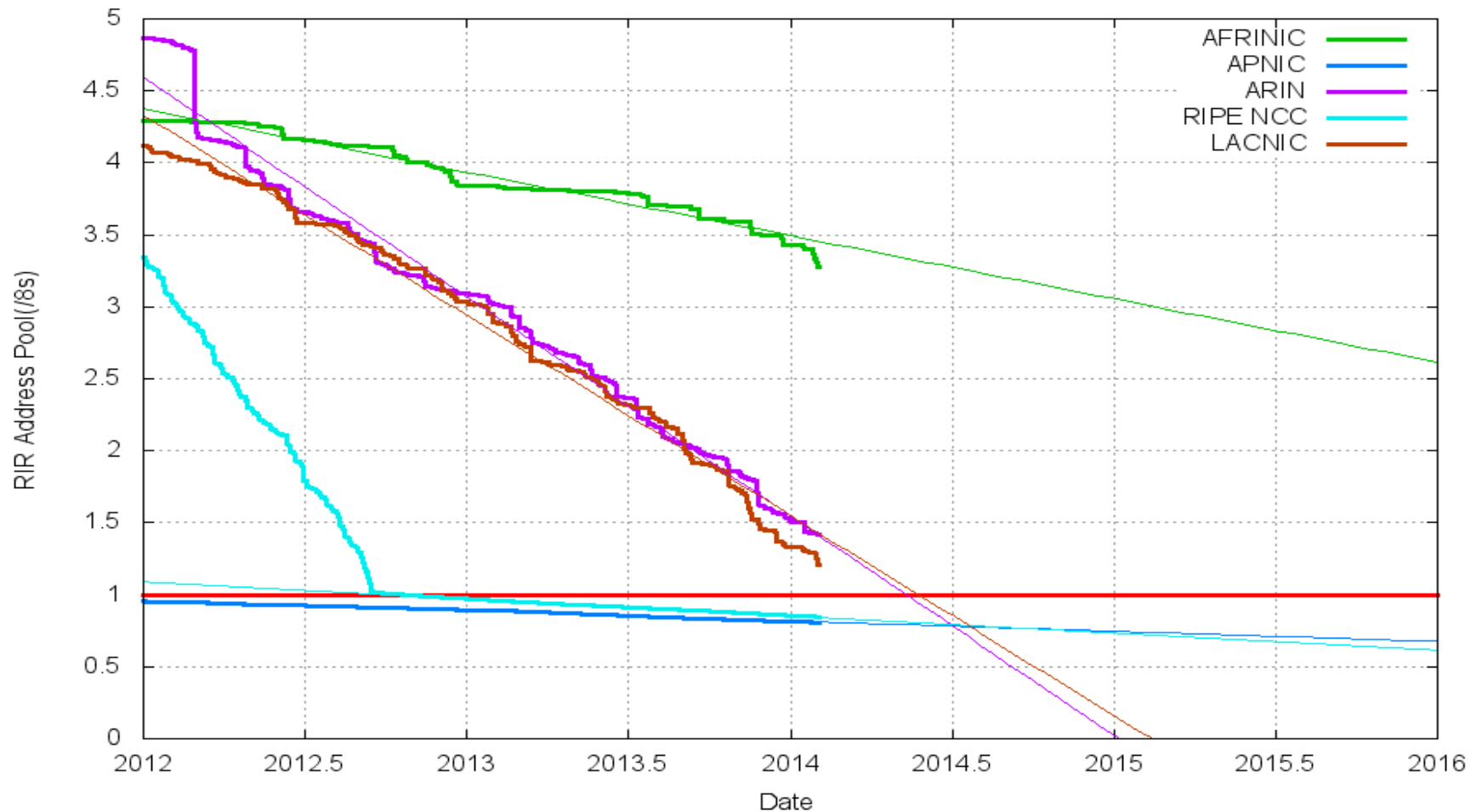
- ◆ Teil I
  - ◆ Warum IPv6
  - ◆ Notation
  - ◆ Features
  - ◆ Aktueller Stand
- ◆ Teil II
  - ◆ Netzplanung
  - ◆ Technische Details
- ◆ IPv6 im Produktivbetrieb

## Es war einmal

- ◆ 1970er: 32 Bit für das Experiment Internet v4
  - ◆ Vint Cerf ist Schuld ;-)
- ◆ 1980er: Das Experiment entkommt
- ◆ 1990er: kommerzielle Verbreitung
- ◆ 2000er: Adressraum geht (jetzt wirklich) zur Neige

# Aktueller Stand IPv4

RIR IPv4 Address Run-Down Model



<http://www.potaroo.net/ispcol/2014-02/addressing2013.html>

## IPv4 Dilemma

- ◆ 2014: „nur“ ca. 700-1000 Mio Hosts
- ◆ Adressraum mit 4 Mrd IPs aber quasi voll
- ◆ Früher: /24 (Class C) zu klein → /16 (Class B)
- ◆ Mit CIDR weniger Verschnitt, aber:
  - ◆ Exponentielles Wachstum der Routingtabellen
- ◆ 2014: ca. 500K Routen

## IPv6 - Der Anfang

- ◆ 1991 IETF startet Suche nach Nachfolger
  - ◆ → IP Next Generation (IPng)
- ◆ 1992 erste Vorschläge und Arbeitsgruppen
- ◆ 1994 Verabschiedung des IPv6-Standards

## IPv6

- ◆ 128 Bit
- ◆  $3,4 * 10^{38}$  Adressen
  - ◆ 340 Sextillionen = 667 Brd IPs/cm<sup>2</sup>
- ◆  $7,9 * 10^{28}$  mal so groß wie das IPv4 Internet
- ◆ Jetzt neu mit Buchstaben!
  - ◆ 2a01:4f8:0:a133:222:4dff:fea5:86d7/64
  - ◆ Hexadezimale Notation

## IPv6 Notation

- ♦ 8 Blöcke á 16 Bit getrennt durch Doppelpunkt
- ♦ Führende Nullen im Block können weggelassen werden:
  - ♦ 2a01:04f8:0d0a:2001:0000:0000:0000:0002
  - ♦ 2a01: 4f8: d0a:2001: 0: 0: 0: 2
- ♦ Komprimierte Schreibweise
  - ♦ **Einmalige** Zusammenfassung aufeinanderfolgender Gruppen von Nullen
  - ♦ 2a01:4f8:d0a:2001:0:0:0:2 → 2a01:4f8:d0a:2001::2
  - ♦ FF02:0:0:0:0:0:0:2 → FF02::2
  - ♦ 0:0:0:0:0:0:0:1 → ::1



## IPv6 Netzmaske/Prefixe

- ◆ z.B. IPv4: 255.255.255.0
- ◆ Unbrauchbar für IPv6:
  - ◆ FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000
- ◆ Nutzung der „/“-Notation bei IPv4
  - ◆ 255.255.255.0 → /24
  - ◆ 255.255.255.192 → /26
- ◆ „/“-Notation bei IPv6
  - ◆ FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000 → /64
- ◆ Prefix-Notation:
  - ◆ 2a01:4f8::/29

## IPv6 Features

- ◆ Automatische IP-Konfiguration (SLAAC)
  - ◆ Seit 2010 auch mit Reverse DNS Server (RFC6106)
  - ◆ außer Windows oder Android
- ◆ kein Broadcast mehr, dafür zwingend Multicast
- ◆ ARP heißt jetzt Neighbor Discovery (NDP)
- ◆ (Server-)systeme erhalten Subnetze statt Einzel-IPs (/64)
- ◆ (vorläufig) ausreichend großer Adressraum
  - ◆ bei /64 pro System =  $2^{32}$  \* IPv4 Internet

## IPv6 Adressen im System

- ◆ Jedes System hat in der Regel mehrere Adressen
- ◆ Link-Local
  - ◆ Prefix FE80::/10
  - ◆ nur auf diesem Link gültig
  - ◆ Erhält jede Netzwerkschnittstelle automatisch
  - ◆ generiert aus MAC-Adresse → EUI-64
  - ◆ 70:54:d2:ab:3f:02 → fe80::7254:d2ff:feab:3f02/64
- ◆ Nötig für Adress-Autokonfiguration, Neighbor-Discovery
- ◆ Werden nicht geroutet

## IPv6 Adressen im System

- ◆ Mehrere globale Adressen möglich:
  - ◆ Via SLAAC: 2a01:4f8:0:a133:7254:d2ff:feab:3f02/64
  - ◆ Via Privacy: 2a01:4f8:0:a133:2c61:18f6:5e72:7425/64
  - ◆ Statisch konfiguriert: 2a01:4f8:0:a133::14:1:25/64

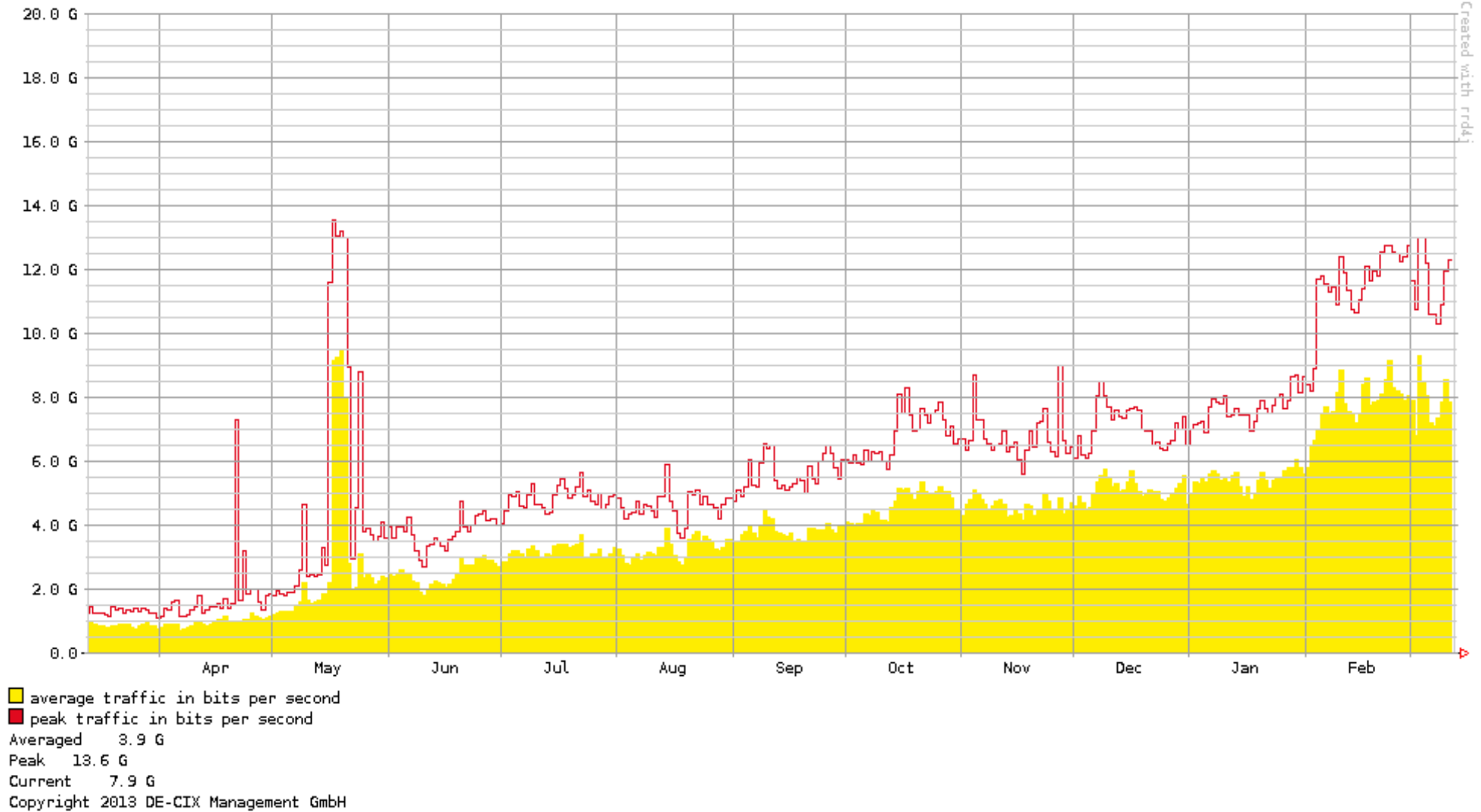
## IPv6 Multicast

- ◆ Zwingend notwendig, da selbst kleinste Netze riesig
- ◆ Prefix: FF00::/8
- ◆ MAC: 33:33:xx:xx:xx:xx (v4: 01:00:5e:[0-7]x:xx:xx:xx )
- ◆ Beispiele
  - ◆ FF02::1 – All Nodes (eines Links)
  - ◆ FF02::2 – All Routers (eines Links)
  - ◆ FF05::1:3 – Alle DHCP Server einer Site

## Routing mit Adresshierarchie

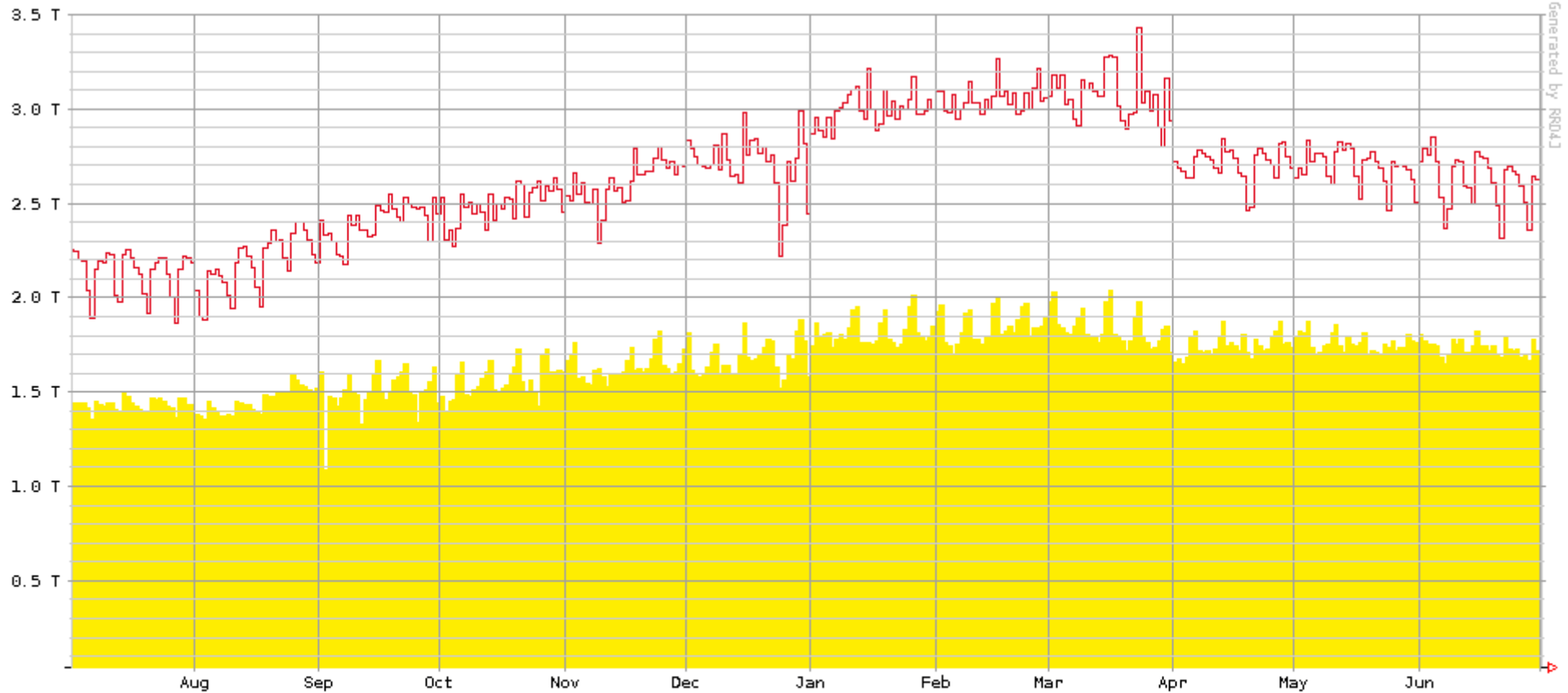
- ♦ Ziel Verkleinerung globale Routingtabelle
  - ♦ Zusammenfassen von Prefixen
- ♦ Adressen mit topologischer/geographischer Bedeutung
  - ♦ IANA – 2000::/3
  - ♦ RIPE (Europa) – 2a00::/12
  - ♦ Hetzner: 2a01:4f8::/29
- ♦ Konsequenz:
  - ♦ Re-Adressierung bei Umzug oder Topologieänderung

## DE-CIX IPv6 2012-2013



WE LOVE BITS. DO YOU?

## DE-CIX IPv4 letztes Jahr



■ average traffic in bits per second

■ peak traffic in bits per second

Current 1717.8 G

Averaged 1662.8 G

Graph Peak 3431.7 G

DE-CIX All-Time Peak 3431.68 G - reached at 2014-03-23T19:00+01:00

Copyright 2014 DE-CIX Management GmbH

# WE LOVE BITS. DO YOU?



## Stand IPv6

- ♦ Juli 2014: weniger als 1% des Traffics IPv6
  - ♦ DE-CIX: 8 Gbit vs 1800 Gbit
- ♦ <http://www.google.com/ipv6/statistics.html>
  - ♦ 4,08% der Zugriffe (2013: 1,19%)
- ♦ Deutschland holt auf: 8,63% (2013: 1,69%)
  - ♦ Frankreich: free.fr – seit 2007 (gesamt >5%)
  - ♦ Schweiz/Rumänien: >10%
  - ♦ Belgien: > 19%

## Warum?

- ◆ Verfügbarkeit (oft abhängig von wenigen Anbietern)
- ◆ genügend(?) große IPv4-Adressreserven
- ◆ NAT, Carrier Grade NAT, NAT444
- ◆ Komplexität, Gleichgültigkeit, Desinteresse
- ◆ never touch a running system
- ◆ Fehlende und fehlerhafte Software
- ◆ Sicherheitsprobleme in/durch IPv6 (\*hust\*)

## Warum nicht?

- ◆ IPv6 ändert aus Anwendersicht nichts
  - ◆ Kein „Killerfeature“
  - ◆ → keine Nachfrage
- ◆ Viele Vorteile im Netzbetrieb
- ◆ IPv4-Wissen ist nicht plötzlich veraltet oder falsch
- ◆ NAT ist veraltet!
- ◆ Sanfter Übergang durch Parallelbetrieb

# Teil II

## Wieviele IPv6 bekommt man so?

- ♦ Anfangs /48 (=65536 /64) pro Site angedacht (RFC3177)
- ♦ auf /56 (=256 /64) pro Site reduziert (RFC6177)
- ♦ /60 pro Kunde (US DSL Anbieter) = 16 /64
- ♦ /64 pro Root-Server
  - ♦ optional /56
- ♦ /48 minimale PI (Provider Independent) Allokation
- ♦ RIPE-Minimum /32 an LIR (auf Wunsch /29)
  - ♦ DTAG 2003::  - ♦ US DoD: 2608::

## Netzplanung

- ◆ Umdenken von IPv4 nötig
- ◆ in Subnetzen (/64) denken!
  - ◆ keine Clientsysteme mehr zählen
- ◆ Adressraum ist groß genug
  - ◆ Allokation sollte für die nächsten 10+ Jahre reichen
- ◆ echte Hierarchie von Anfang an planen
  - ◆ keine Netzfragmentierung
- ◆ Best current practice
  - ◆ Subnetting an Nibble (4 Bit) = Hex-Stelle der Adresse
  - ◆ Aber: bis zu 16x zu viele Adressen

## Netzplanung (cont.)

- ◆ Beispiel: /64 pro Server oder Client-Subnet
  - ◆ 24/48-Port: /56 statt /59 bzw /58
  - ◆ Uplink 12-Port Router: /52 statt /55 bzw. /54
  
- ◆ Beispiel: /48 als Site-Allokation oder PI
  - ◆ 10 (Firmen-)standorte: Minimum 4 Bits ( $2^4 = 16$ )
  - ◆ = /52 pro Standort = 4096 /64 Subnetze
  - ◆ oder /53 pro Standort = 2048 /64

## Netzplanung (cont.)

- ◆ Prefixe kleiner als /64 möglich
  - ◆ aber ohne „Bling“ wie SLAAC, DAD
  - ◆ z.B. für VMs oder NAT64 (/96)
- ◆ [http://www.ripe.net/lir-services/training/material/IPv6-for-LIRs-Training-Course/IPv6\\_addr\\_plan4.pdf](http://www.ripe.net/lir-services/training/material/IPv6-for-LIRs-Training-Course/IPv6_addr_plan4.pdf)



## **Anschlüssen nicht vergessen!**

- ◆ IPv6 ist fast überall per default aktiviert und wird bevorzugt
- ◆ Neighbor Discovery (ND) anfällig gegen Spoofing wie ARP
  - ◆ Secure ND schwierig (erfordert PKI)
- ◆ SLAAC nur für vertrauenswürdige Netze !!!11elf
  - ◆ IPv6 wird gegenüber v4 bevorzugt (MITM)
  - ◆ Fake RA / RA Flooding
  - ◆ Fake DAD

## **Anschlüssen nicht vergessen!**

- ◆ Kundennetze auf Link-Local-IP routen
  - ◆ statisches ND (aka static ARP für v4)
  - ◆ verhindert Flooding mit vielen NDP Einträgen
- ◆ Server in öffentlichen Netzen sollten v6-Magie abschalten
  - ◆ autoconf
  - ◆ accept\_ra\*
  - ◆ accept\_dad aus

## Hetzner IPv6 Status für Kunden

- ♦ /64 kostenlos verfügbar für Kunden seit Mitte 2010
  - ♦ optional /56
- ♦ vorkonfiguriert in allen Linux-Standardinstallation
- ♦ Reverse DNS Einträge
- ♦ Monitoring
- ♦ DNS Resolver
- ♦ Authoritative Nameserver

## IPv6 Migration

- ◆ schrittweise Umstellung
- ◆ Testphase mit v6-only und Dual-Stack-Subdomain
- ◆ DNS Resolver, Backupserver, Mirror, NTP
- ◆ nach IPv6 Launch-Day (Juni 2012)
  - ◆ sukzessive Umstellung auf echtes Dual-Stack
- ◆ Umstellung auf geroutete Kundennetze
  - ◆ fe80::1 als Default-GW
  - ◆ war erst nach Update der Router-Firmware durch Hersteller möglich

## **IPv6 TODO / Wishlist**

- ◆ Delegation rDNS
- ◆ /56 als Standard
- ◆ v6-only Server

## Lessons learned IPv6

- ◆ nur noch wenig (Server)software mit IPv6-Problemen
  - ◆ z.B. Debian 6.0.x RPC/NFS-Server
- ◆ Inzwischen RIPE Vorgaben zur Vergabe / Netzdesign
  - ◆ erlauben „schöneres“ Subnetting – an den Nibbles ;-)
  - ◆ Policy für Beantragung von neuen Netzen
- ◆ v6-only ist keine Option
  - ◆ außer man braucht nur Google, Facebook und Youtube
  - ◆ fehlender v6-Support in Client-Software:
    - ◆ z.B. Skype, Google Hangouts, u.v.a. (Android) Apps
    - ◆ <https://sites.google.com/site/tmoipv6/464xlat>

# Deployment

## **ABLAUF OS DEPLOYMENT**

- ◆ Start des Servers via Wake on LAN
- ◆ PXE Boot des Installers
- ◆ Installation via Kickstart / FAI / etc.
- ◆ Post-Install Skripte



Legacy (BIOS) PXE kann kein IPv6 und wird es  
auch nie können

„Intel is not investing any time or resources on improvements to legacy PXE implementations for client systems. All our present and future energies are focused on UEFI Spec network boot implementation (which covers both IPv4 and IPv6).“

## Und nun?

- ◆ Egal.
- ◆ Deployment via Legacy PXE
  - ◆ v4 bootstrap
  - ◆ v6 Konfiguration während/nach der Installation
- ◆ Problem für ISPs / Hoster
  - ◆ jedes System braucht auf mind. 1 öffentliche IPv4
  - ◆ Alternativ Nutzung von RFC1918/RFC6598 Adressen
- ◆ Aber
  - ◆ Windows-Installation auf HDDs > 2TiB
  - ◆ Nur im UEFI Modus möglich

## UEFI?

- ◆ Unified Extensible Firmware Interface
  - ◆ wird BIOS ablösen
  - ◆ Bootloader in eigener Partition (EFI System Partition)
- ◆ erste UEFI Boards seit 2011
- ◆ PXE zunächst nur im Legacy BIOS Modus
  - ◆ kein Zugriff auf UEFI Variablen
- ◆ Seit Mitte 2012 auch Mainboards UEFI PXE
  - ◆ Aber pxelinux.0 ist kein EFI-Binary

## Dual BIOS/UEFI INFRASTRUKTUR

- ◆ DHCP Server muß unterscheiden können
  - ◆ BIOS PXE Client
  - ◆ UEFI Netboot Client
- ◆ Bootfile / Bootmenü
  - ◆ BIOS: PXELINUX
  - ◆ UEFI: z.Z. nur durch Redhat EFI-GRUB1
- ◆ Nachteil:
  - ◆ Pflege von mehreren Systemen
  - ◆ Windowsinstallation mit Linux-Boardmitteln nur mit Tricks möglich

- ◆ Wenn man IPv6 bekommen kann, anfangen!
- ◆ Testen, Ausprobieren, Lernen
- ◆ Deployment vorerst über IPv4
- ◆ UEFI ist die nächste Großbaustelle

Fragen? Fragen!

**WE LOVE BITS. DO YOU?**

## **Wir suchen Mitarbeiter!**

- ♦ 1. Ausbildung Fachinformatiker
- ♦ 2. System und Netzwerk-Administratoren
- ♦ 3. Abschlußarbeiten / Praktika / Ferienarbeit

**mehr unter *<https://jobs.hetzner.de>***

## DAS UNTERNEHMEN

**HETZNER ROOT SERVER**

**HETZNER ONLINE**

### SEHR GEFRAGT!

*Zuverlässiger und preiswerter Root Server sucht anspruchsvollen User!*

*Bringe mit Vollen Root-Zugriff, viel Power, maximale Verfügbarkeit und hohe Effizienz*

[www.hetzner.de](http://www.hetzner.de)

HETZNER ROOT SERVER EX 4	HETZNER ROOT SERVER EX 5
• Intel®Core™ i7-2600 Quad-Core inkl. Hyper-Threading-Technologie	• Intel®Core™ i7-920 Quad-Core inkl. Hyper-Threading-Technologie
• 16 GB DDR3 RAM	• 24 GB DDR3 RAM
• 2 x 3 TB SATA 6 Gb/s HDD 7200 rpm (Software-RAID 1)	• 2 x 750 GB SATA 3 Gb/s HDD (Software-RAID 1)
• Linux-Betriebssystem	• Linux-Betriebssystem
• Traffic enthalten*	• Traffic enthalten*
• IPv6-Subnetz (1/64)	• IPv6-Subnetz (1/64)
• Domain-Registration-Roboter	• Domain-Registration-Roboter
• Keine Mindestvertragslaufzeit	• Keine Mindestvertragslaufzeit
• Setupgebühr 49 €	• Setupgebühr 0 €
monatlich <b>49€</b>	monatlich <b>59€</b>

**GreenIT Best Practice Award 2011**

Hetzner Online unterstützt mit der Verwendung von 100% regenerativem Strom aktiv den Umweltschutz. Entdecken Sie sich gemeinsam mit uns für eine bessere Zukunft.

[WWW.HETZNER.DE](http://WWW.HETZNER.DE)

\* Der Trafficverbrauch ist kostenlos. Bei einer Überschreitung von 1000 GB/Monat wird die Abrechnung auf 10 MBps reduziert. Optional kann für 4,9€ ein weiteres TB an Bandbreite dazugelassen (auf 100 MBps begrenzt werden).

Hetzner Online ist ein professioneller Webhosting-Dienstleister und erfahrener Rechenzentrenbetreiber. Wir bieten Lösungen an, die durch Qualität, Stand der Technik und Sicherheit überzeugen. Dabei reicht das Angebot für Homepage-Einsteiger bis zum professionellem Webentwickler:

- ◆ Root, Managed und vServer
- ◆ Colocation
- ◆ Shared Hosting
- ◆ Internet Domains
- ◆ SSL-Zertifikate