

# IPv6-Überblick

Gert Döring

30. September 2003 / 21. Oktober 2003

## Übersicht

- IPv6: Was und warum?
- IPv6-Adressen und Adress-Hierarchie
- DNS
- Konfigurationsbeispiele
- Routing: Protokolle und Zahlen
- Multicast
- Migration IPv4/IPv6
- offene Probleme

## IPv6: Erste Eindrücke

- Adressen sehen anders aus: 2001:608:0:1::100
- Traceroute sieht anders aus:

```
gert@tuerkis:/home/gert$ traceroute6 -l turing3.space.net
traceroute6 to turing3.space.net (2001:608::1000:44) from 2001:608:4::2,
30 hops max, 12 byte packets
 1  cisco.greenie.muc.de (2001:608:4::1)  5.196 ms  4.748 ms  *
 2  Cisco-M-VIII.space.net (2001:608::119)  46.282 ms  34.808 ms  38.9 ms
 3  turing3.space.net (2001:608::1000:44)  37.908 ms  37.104 ms  56.227 ms
```

- Aber: Applikationen funktionieren “ganz normal”:

```
gert@moebius2.space.net:/home/gert$ telnet m8
Trying 2001:608::119...
Connected to Cisco-M-VIII.space.net.
Escape character is '^]'.

Username:
```

## IPv6: Warum will man das?

- IPv4-Adressknappheit (\*) führt zu:
  - dynamischen IPs für Einwahl/DSL
  - NAT (double-NAT, VPN-Adresskollisionen, ...)
  - Problemen mit End-to-End-Applikationen (VoIP, PtP)
  - Kollision zwischen “Security” und “Conservation” (Hosting)
  - Ärger zwischen ISPs und Kunden (“ich will ein C-Netz!”)
- Gesamtzahl an IPv6-Adressen:  $2^{128}$ , d.h.,  
340'282'366'920'938'463'463'374'607'431'768'211'456 IPs
- $\Rightarrow$  umgerechnet: über 1000 **Netze** (/48) pro Person
- $\Rightarrow$  Adressknappheit ist mit IPv6 komplett vom Tisch
- Bedarf seitens UMTS/GPRS-Netzen, China, Japan, usw.

## IPv6: Warum wollte man das?

- Mit “IP next generation” (IPng) wurden neue Mechanismen entwickelt, die mit den Problemen von IPv4 aufräumen sollen
- IP-Adreßknappheit → OK
- Security: IPSEC → auch für IPv4
- IP-Autoconfig → auch für IPv4 (DHCP)
- Mobility → fraglich, gibt’s auch bei v4, aber v6 ist eleganter
- Performance-Optimierung (IP-Header besser strukturiert, einfacher erweiterbar und trotzdem für Router schnell “in Hardware” zu bearbeiten) → durch bessere Router-Hardware nicht mehr zwingend erforderlich
- Routingtable-Wachstum → hilft nur begrenzt

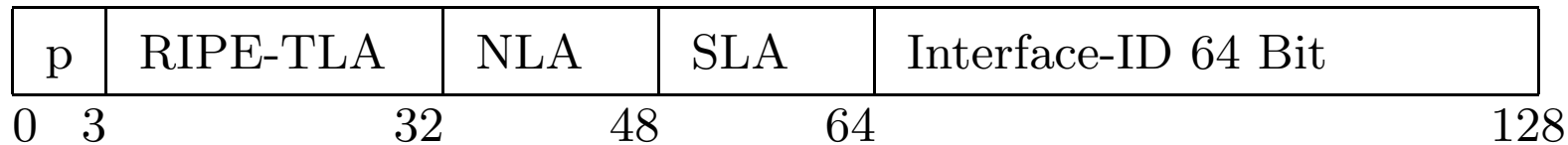
## IPv6: Adressen näher betrachtet

- IPv4-Adressen sind 32 Bit lang, werden als 4x 8 Bit dezimal (0-255) geschrieben: 195.30.0.1
- IPv6-Adressen sind 128 Bit lang, und werden als 8x 16 Bit hexadezimal (0000-ffff) geschrieben:  
2001:0608:0000:0001:0000:0000:0000:0100
- führende Nullen können weggelassen werden:  
2001:608:0:1:0:0:0:100
- *ein* Block aus zusammenhängenden Nullen kann zu ::  
zusammengefaßt werden (muß eindeutig sein!):  
2001:608:0:1::100

## IPv6: Adressen, die man so sieht

- Loopback: `::1` (vgl: IPv4: 127.0.0.1)
- 6Bone-Test: `3ffe:...` ([www.6bone.net](http://www.6bone.net))
- RIPE: `2001:0600:...` ... `2001:09ff:...`
- ARIN: `2001:04xx:...`
- APNIC: `2001:02xx:...`, `2001:0axx:...`
- Link-Local: `fe80::...` (vgl: IPv4: RFC1918)
- Site-Local: `fec0::...` (NEU!) - *deprecated*
- 6to4: `2002:xxx:xxx:...`
- IPv4-mapped: `::FFFF:<ipv4>` (nur an Socket-API)

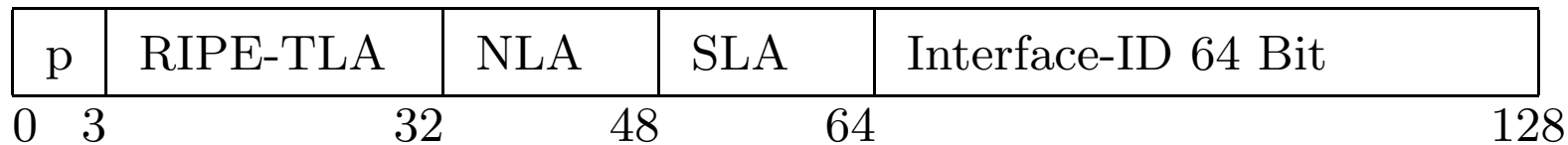
## IPv6: Adress-Hierarchie und RIPE



- CIDR-Notation: “Netzadresse/Bits”, 2001:608::/32
- derzeit nur p=001 vergeben
- Subnets sind **immer** /64s  
(auf Multiaccess-Medien)
- Kunden bekommen **immer** ein /64 oder ein /48 (!)
- Provider bekommen von RIPE ein /32 (alt: /35) oder mehr
- Reseller bekommen vom ISP ein /33.../46, je nach Bedarf



## IPv6: Adress-Hierarchie bei Kunden



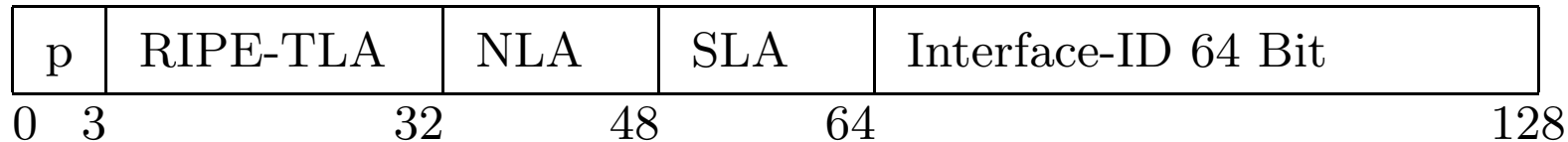
- Kunden mit Subnetzen bekommen **immer** ein /48
- LAN-Segmente sind **immer** ein /64
- 16 Bit SLA-Space → 65536 Subnetze, oder eine interne Hierarchie mit (z.B.) 256 Standorten (8 Bit) und dort jeweils 256 Netzsegmenten
- Bei Umzügen zwischen Providern kann die Adreßstruktur beibehalten werden, nur das /48-Prefix muß angepasst werden
- Site-Local-Prefix `fec0::/48` für geschlossene Netze  
(*deprecated* nach `draft-ietf-ipv6-deprecate-site-local-00.txt`, neue Ansätze in Arbeit)

## IPv6: Endgerätekonfiguration (1)

p	RIPE-TLA	NLA	SLA	Interface-ID 64 Bit
0 3	32	48	64	128

- Stateless Autoconfig in Multiaccess-Netzwerken: Prefix /64 wird extern festgelegt, aus MAC-Adresse wird die Gesamt-IP-Adresse automatisch errechnet (EUI-64, RFC 3513, 2.5.1)
- Es gibt **immer** fe80::/64 (link-local)
- optional: Announcement von Site-Local (fec0:0:0:xxxx::/64) und/oder ISP-Prefixen (2001:608:1:0::/64) via *Router-Announcements* (RAs, RFC 2461).  
Hosts hängen daran die EUI-64-Adresse als *host part*.
- → Rechner haben fast immer mehrere IPv6-Adressen
- Prefix mit Lifetime announced → einfacheres Renumbering

## IPv6: Endgerätekonfiguration (2)



- statische Konfiguration möglich und wünschenswert (für Server), auch Mischbetrieb möglich
- Beispiel: ns6.dns.space.net:  

```
ifconfig -a  
  ether 00:80:c8:56:bf:25  
  inet6 fe80::280:c8ff:fe56:bf25%de0 prefixlen 64 scopeid 0x1  
  inet6 2001:608::280:c8ff:fe56:bf25 prefixlen 64  
  inet6 2001:608::1000:1 prefixlen 64
```
- “privacy extentions” für Datenschützer - RFC3041
- Problempunkt Autokonfiguration: DNS-Updates, Accounting

## DNS mit IPv6

- Forward-DNS: über neuen Resource-Record “AAAA”

```
miyu      IN      A        193.149.48.180
          IN      AAAA     2001:608:4:0:2c0:26ff:feee:91ea
          IN      MX      100      kirk
```

- Reverse-DNS: umgedreht und Punkte an Nibble-Grenzen:

```
2001:608:4:0:2c0:26ff:feee:91ea
2001:0608:0004:0000:02c0:26ff:feee:91ea
ae19 eeef ff62 0c20 0000 4000 8060 1002
a.e.1.9. e.e.e.f. f.f.6.2. 0.c.2.0. 0.0.0.0. 4.0.0.0.
      8.0.6.0. 1.0.0.2. ip6.arpa. IN PTR miyu.greenie.muc.de.
```

- prinzipiell analog IPv4:

```
180.48.149.193.in-addr.arpa.      IN PTR miyu.greenie.muc.de.
```

## DNS mit IPv6(2)

- Reverse-DNS von Hand: fehleranfällig
- Forward-DNS umständlich, wenn man ganze Netze umzieht (also eigentlich nur überall ein neues /48- o. /64-Prefix dazukommt und später das alte wegfällt)
- Idee zur Abhilfe: A6 und DNAME-Records mit Referenzen auf andere Records (“nimm prefix.space.net und häng’ ::1000:1 an”). Wegen zu erwartender Probleme mit der Stabilität des DNS wieder eingestampft.
- Lösung muß bessere Verwaltung der DNS-Zonen im BIND sein
- offen ist auch noch IPv6-Autoconfig → DNS
- Historisches Hindernis beim Reverse-DNS: ip6.int → ip6.arpa

## Konfiguration: Cisco

- braucht 12.2(x)T / 12.3(x) IOS oder aktuelle BETAs

- Global:

```
m8(config)# ipv6 unicast-routing
```

- Interfaces:

```
m8(config)# interface eth1
```

```
m8(config-if)# ipv6 addr 2001:608:0:1::1/64
```

```
m8(config)# interface bri1:1
```

```
m8(config-if)# ipv6 unnumbered eth0
```

- Routen:

```
m8(config)# ipv6 route 2001:608:0:1003::/64 BRI1:1
```

- Kontrolle:

```
m8# show ipv6 ? / ping / traceroute
```

## Konfiguration: FreeBSD/NetBSD

- braucht FreeBSD 4.1, NetBSD 1.5 oder später

- Konfiguration über /etc/rc.conf:

```
ipv6_enable="YES"
```

```
ipv6_ifconfig_de0="2001:608::2 prefixlen 64"
```

```
ipv6_ifconfig_de0_alias0="2001:608::1000:2 prefixlen 64"
```

```
ipv6_defaultrouter="2001:608::1"
```

- Konfiguration von Hand:

```
# ifconfig de0 inet6 add 2001:608:0:777::123 [prefixlen 64]
```

- Routen:

```
# route add 0::0/0 2001:608::1
```

- Kontrolle:

```
# netstat -an / netstat -rn / ping6 / traceroute6
```

## Konfiguration: (RedHat) Linux (1)

- braucht Kernel 2.2.x oder 2.4.x mit IPv6
  - ggf. IPv6-Modul laden: `modprobe ipv6`
  - Globale Aktivierung: `/etc/sysconfig/network`
- Konfiguration über `.../network-scripts/ifcfg-eth0`

```
IPV6INIT=yes
```

```
IPV6ADDR=2001:608:0:8::130[/64]
```

```
IPV6_DEFAULTGW=2001:608:0:8::1
```

```
IPV6_AUTOCONF=no
```

```
IPV6_ROUTER=no
```

```
IPV6FORWARDING=no
```

- Kontrolle:

```
# netstat -an / netstat -rn -A inet6 / ping6 / traceroute6
```



## Konfiguration: (RedHat) Linux (2)

- Konfiguration von Hand:

```
# ifconfig eth0 inet6 add 2001:608:0:777::123[/64]
```

- Routen:

```
# route -A inet6 add 2000::/3 gw 2001:608::1 [dev eth0]
```

```
# ip -6 route add 2000::/3 via 2001:608::1 [dev eth0] metric 1
```

- Caveat: Linux-Router wollen 2000::/3 als Default-Route
- Caveat: Auto-Configuration immer aktiv (?) - ggf. auf dem Router deaktivieren (ip6 nd suppress-ra oder ip6 nd prefix default no-autoconfig)

## Konfiguration: Windows XP

- bei XP eingebaut, bei NT/2000 über download
- Global einschalten (im “cmd”):  
`ipv6 install`
- Kontrolle:  
`ipv6 if / ping6`
- <http://www.microsoft.com/windowsxp/pro/techinfo/administration/ipv6/default.asp>

## Programmierinfos (Unix-Sockets)

- die meisten Programme, die über TCP- oder UDP-Sockets kommunizieren, lassen sich sehr einfach anpassen:

alt		neu
gethostbyname()	→	getaddrinfo()
gethostbyaddr()	→	getnameinfo()
ntoa(), printf(“%d.%d.%d.%d”)	→	inet_ntop()

- mühsam sind Applikationen, die auf IP-Basis Zugriffskontrolle machen - Vergleichsfunktionen müssen angepasst werden.
- knifflig sind Protokolle, die IP-Adressen im Protokoll selbst verwenden (FTP/Port und IRC/DCC) → Protokollerweiterung nötig (FTP: EPSV), und Applikationen, die auf Paket-Ebene arbeiten (tcpdump, BIND, traceroute, ...)

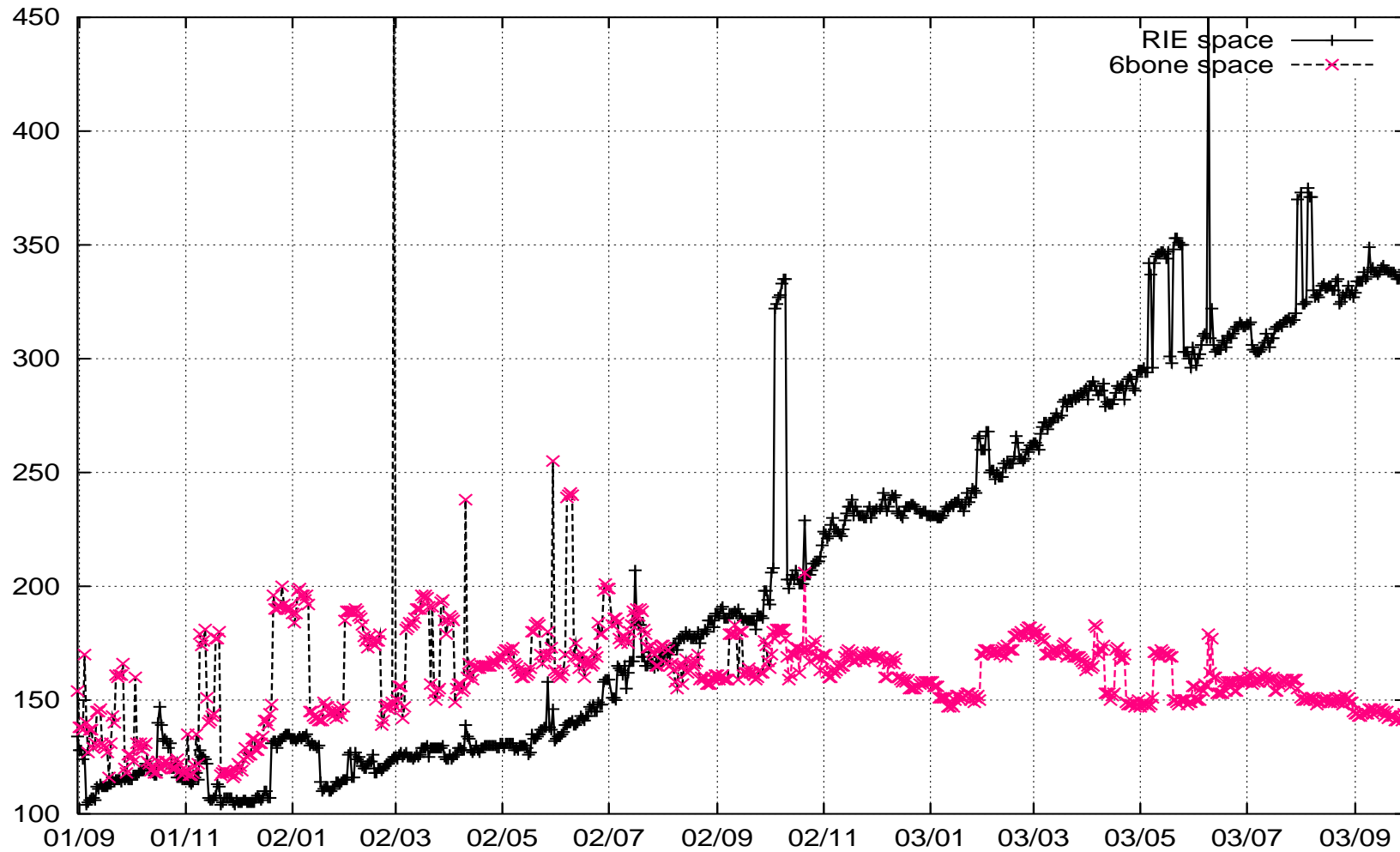
## IPv6 Routing: Protokolle

- Unicast-Routing “wie immer”: most specific wins, VLSM
- BGPv4++ - RFC2545, RFC2858
  - BGPv4 mit Multiprotokoll-Erweiterung
  - IPv4 unicast / multicast, IPv6 unicast / multicast, ...
- RIPng - RFC2080
  - analog RIPv2: distance-vector, multicast-based
- ISIS - draft-ietf-isis-ipv6-05.txt, analog RFC1195
  - minimale Erweiterung, integriert ISO, IPv4, IPv6
- OSPFv3 - RFC2740
  - neues Protokoll für IPv6 only (theoretisch auch MP)
  - IPv4 legacy z.B. in 32-Bit-Router-ID

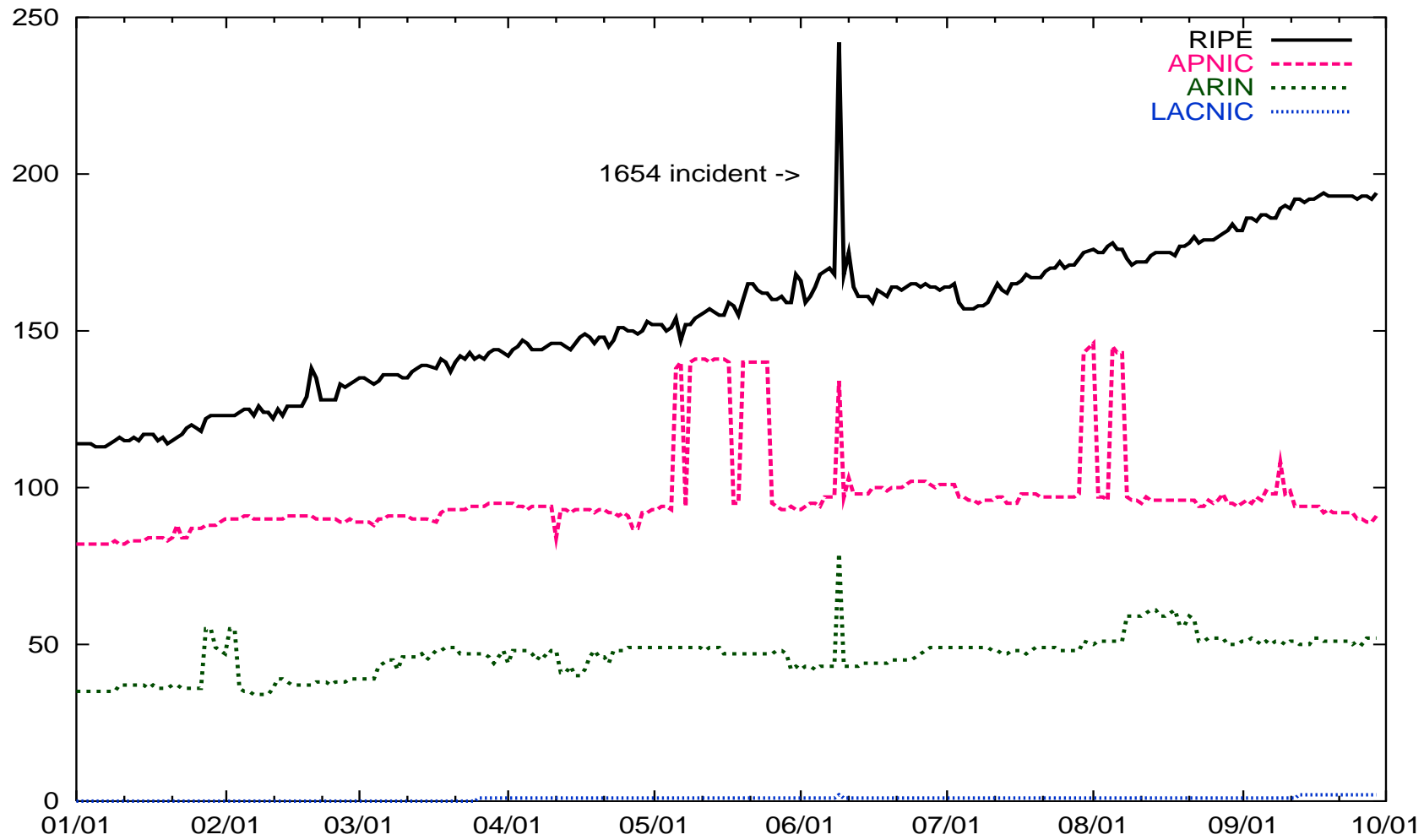
## IPv6 Routing: BGP4 packet extensions

- TCP-Session über IPv4 oder IPv6 transport (!)
- IPv6-Prefixe in neuem Attribut MP\_REACH\_NLRI:
  - + Frame 9 (200 bytes on wire, 200 bytes captured)
  - + Ethernet II, Src: 00:b0:8e:91:38:1a, Dst: 00:00:0c:3d:dc:c9
  - Internet Protocol Version 6
    - Source address: 2001:608:0:10::115 (2001:608:0:10::115)
    - Destination address: 2001:608:0:10::99 (2001:608:0:10::99)
    - Next header: TCP (0x06)
  - + Transmission Control Protocol, Src Port: 179, Dst Port: 39463, Len: 126
  - Border Gateway Protocol
    - UPDATE Message
      - Total path attribute length: 65 bytes
      - Path attributes
        - ORIGIN: IGP (4 bytes)
        - AS\_PATH: empty (3 bytes)
        - MULTI\_EXIT\_DISC: 0 (7 bytes)
        - LOCAL\_PREF: 100 (7 bytes)
        - COMMUNITIES: 0:100 0:1000 5539:500 (15 bytes)
      - > MP\_REACH\_NLRI (29 bytes)
        - > Address family: IPv6 (2)
        - > Subsequent address family identifier: Unicast (1)
        - > Next hop network address (16 bytes)
          - > Next hop: 2001:608:0:10::115 (16)
        - > Subnetwork points of attachment: 0
        - > Network layer reachability information (5 bytes)
          - > 2001:608::/32

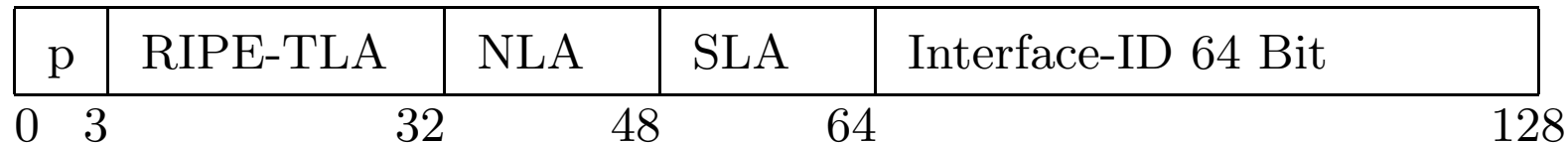
## BGP-Table: RIR (2001) vs. 6Bone (3ffe)



## BGP-Table: Prefixe pro RIR



## IPv6-Multicast



- IPv6-Multicast-Adressen sind **FFfs:...**
- p = 111 (!), f = flag, s = scope
- Link-Local-Multicast **ff02:...** wird für Neighbor-Discovery (= ARP), Router-Announcements (**ff02::1**), usw. benutzt
- Neighbor-Discovery: Multicast auf **ff02::1:ff<low 24bit>**
- Es gibt keine Broadcasts mehr (!) → alles Multicasts
- Global Scope: **ff0e:...** (wie 224.\*), kaum Apps
- Referenz: RFC2375, RFC3306, RFC3307, RFC3513



## IPv6 multicast routing / forwarding

- PIM6 SM/DM (KAME: pim6sd, pim6dd)
- M-BGP mit “address-family ipv6 multicast”
- *kein* MSDP für IPv6
- → embedded RP proposal: Multicast-Gruppen-Adresse enthält Adresse des RPs, zu dem gejoined werden soll.  
draft-savola-mboned-mcast-rpaddr-03.txt
- → oder source specific multicast (SSM, RFC3569, RFC3306):  
alle Joins gehen zu (S,G), nie zu (\*,G) - damit entfällt  
Notwendigkeit für MSDP

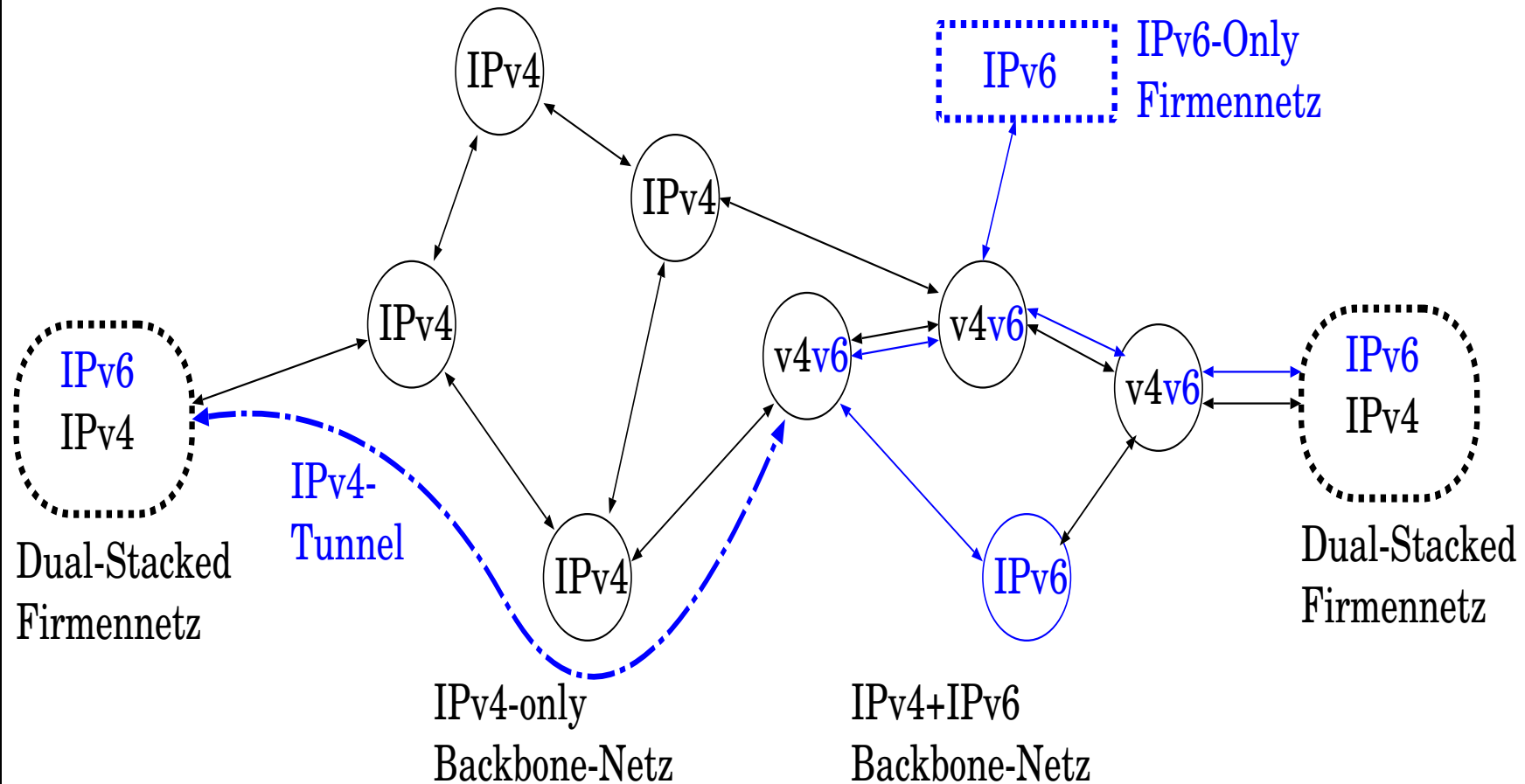
## IPv6: ICMP

- Grundprinzip unverändert (type/code)
  - error messages: *type* <128, informational: *type* >=128
- analog v4: Echo Request/Reply
- analog v4: Dest. Unreach., Time Exc., Param. Prb., Redirect
- neu dazu: Neighbor Solicitation/Advertisement (früher: ARP)
- neu dazu: Router Solicitation/Advertisement
- überarbeitet: Packet Too Big (war: frag. req.)
- weggefallen: Source Quench, Time-Stamp, Address-Mask, Info-Request, div. Dest.-unreach codes
- TTL255-Trick gegen Fälschungen von NS/RS/redirects
- RFC2461 (ND/RD), RFC2463 (ICMPv6)

## Migration: Host-Migration

- Problemstellung: Kommunikation zwischen v4- und v6-Hosts
- Dual-Stack-Hosts (v4+v6)
- NAT/PT am Border-Router - intern v6-only, extern v4-only
- Dual-Stack-Gateways (z.B. Squid, dnscache mit v4/v6)
- Transparente Proxies (FreeBSD faithd)
- auto tunneling (::<ipv4>) - v6 apps, nur v4 Infrastruktur  
(*deprecated*)
- Teredo: IPv6 über IPv4-UDP, für Hosts hinter v4-only + NAT, in Verbindung mit externem Server/Relay für Erstkontakt, Traffic dann direkt, draft-huitema-v6ops-teredo-00.txt

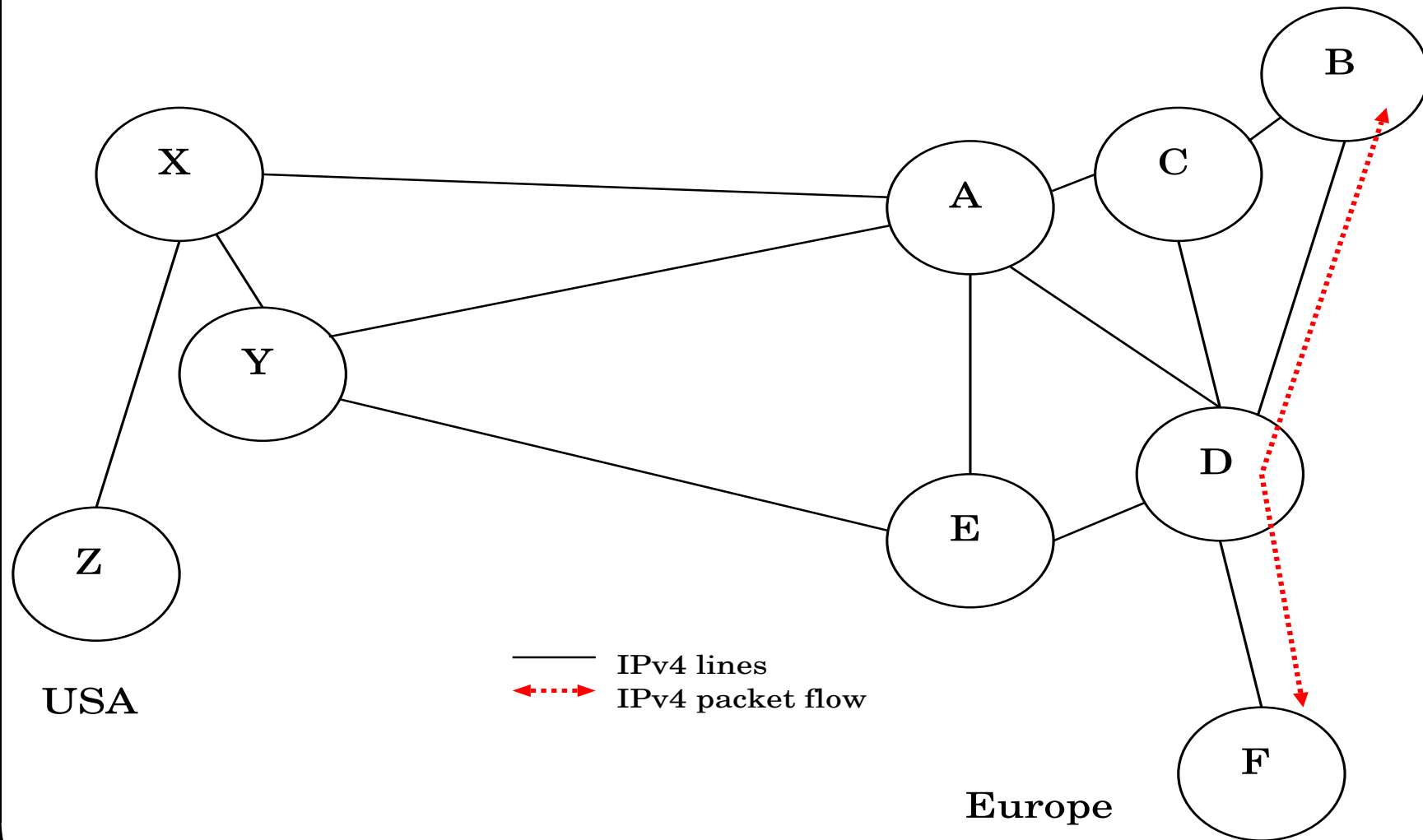
## Migration: v6-LAN an v6-Backbone



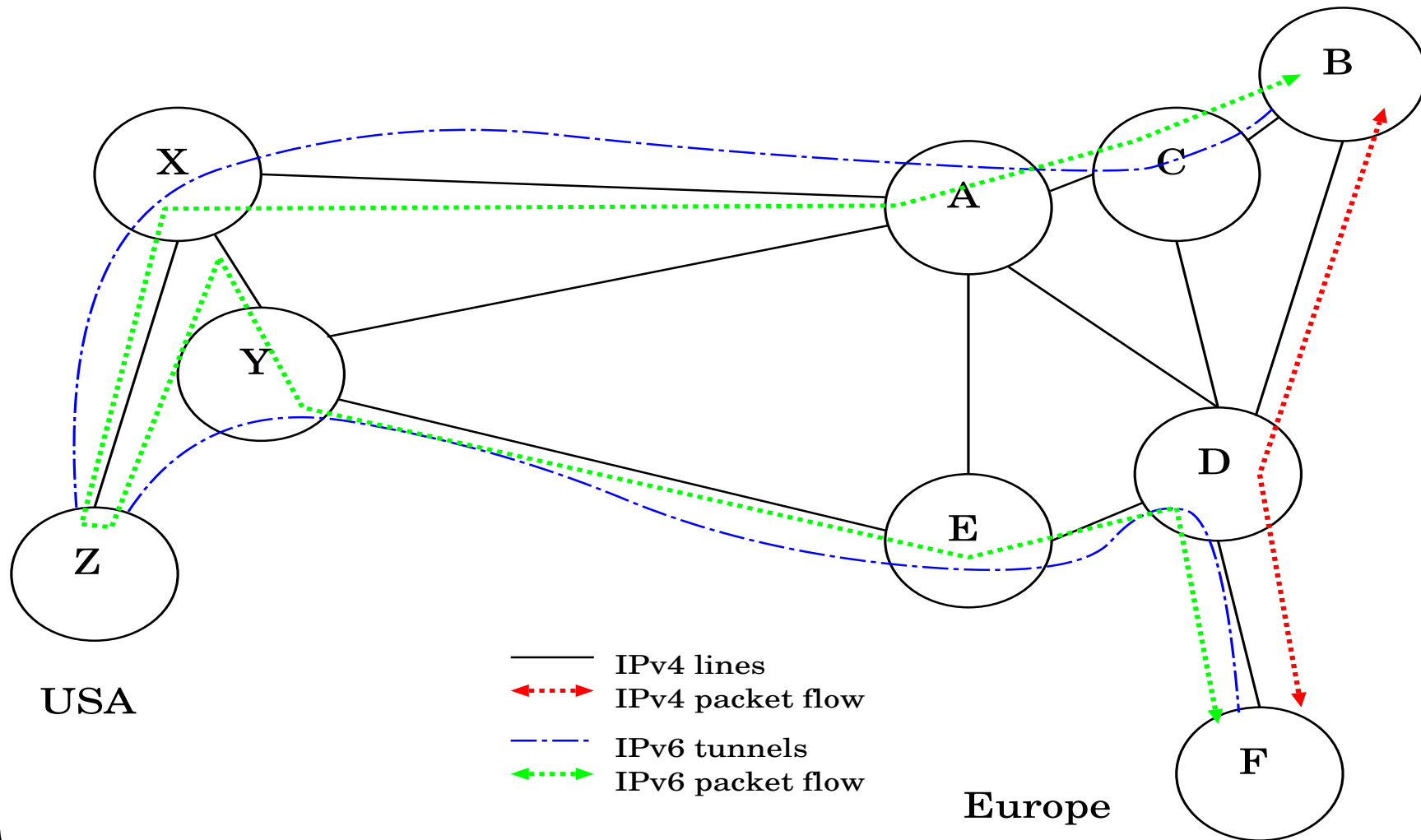
## Migration: Netz-Migration

- Problemstellung: wie verbinde ich mein IPv6-Netz mit anderen IPv6-Netzen, wenn die Netze “dazwischen” nur IPv4 können?
- explizite IPv6-in-IPv4 Tunnel (Cisco mit ip6ip oder gre, Linux mit sit, FreeBSD mit gif, z.B. von SpaceNet zu JW)
- Tunnel als Layer 2 bilden *IPv6 overlay network*
- halbautomatisch: Tunnel Broker, PPTP, L2TP, usw.
- 6to4 - `2002:<router-ipv4>::/48` pro Site, Gateway (u.a.) bei Cisco, Microsoft, DFN
- 6over4 - IPv6 getunnelt über IPv4 multicast, einzelne v6-Hosts über (multicastfähige!) v4-Router
- ISATAP - über IPv4, zentraler Router koordiniert Tunnels

## Tunneling (1): normales BGP/IPv4-Setup



## Tunneling (2): IPv6 über (suboptimale) Tunnel



## Tunneling (3): Beispiel aus der Praxis

```
traceroute to moebius2.space.net (195.30.1.100), 30 hops max, 38 byte packets
 1  213.150.184.129 (213.150.184.129)  4.404 ms  2.820 ms  2.793 ms
 2  172.16.0.1 (172.16.0.1)  7.230 ms  4.513 ms  4.052 ms
 3  193.95.43.99 (193.95.43.99)  4.290 ms  4.840 ms  4.535 ms
 4  193.95.1.38 (193.95.1.38)  5.073 ms  5.237 ms  5.922 ms
 5  193.95.50.202 (193.95.50.202)  6.776 ms  6.549 ms  5.535 ms
 6  pall-tunisie-telecom-9-tn.seabone.net (195.22.205.137)  11.543 ms  11.621 ms  12.611 ms
 7  decix-fra2-racc1.fra.seabone.net (195.22.211.205)  39.658 ms  40.669 ms  42.382 ms
 8  decix.Space.net (80.81.192.105)  39.890 ms  39.544 ms  39.796 ms
 9  Cisco-F-V-Fa6-0-0.Space.Net (193.149.44.254)  41.516 ms  41.866 ms  42.858 ms
10  Cisco-M-XII-POS4-0.Space.Net (195.30.3.33)  44.635 ms  42.800 ms  43.279 ms
11  Cisco-M-XI-Vlan1.Space.Net (195.30.0.116)  49.897 ms
12  moebius2.Space.Net (195.30.1.100)  47.008 ms  47.636 ms  47.945 ms
```

```
traceroute6 to moebius2.space.net (2001:608:0:1::100) from
2001:970:6800:0:202:2dff:fe03:e156, 30 hops max, 16 byte packets
 1  2001:970:6800::1 (2001:970:6800::1)  3.115 ms *  2.926 ms
 2  2001:970:0:2::20 (2001:970:0:2::20)  9.085 ms *  18.411 ms
 3  tu-viagenie.ipv6.noris.de (2001:780::b)  157.448 ms  159.842 ms  155.208 ms
 4  3ffe:b00:c18::11 (3ffe:b00:c18::11)  307.153 ms  336.514 ms  315.435 ms
 5  Cisco-M-VI-Tu18.Space.Net (2001:608:0:3::6)  375.616 ms  358.7 ms  358.521 ms
 6  Cisco-M-VIII.space.net (2001:608::119)  362.091 ms  365.437 ms  361.65 ms
 7  moebius2 (2001:608:0:1::100)  359.427 ms  368.408 ms  359.534 ms
```

⇒ IPv6 geht zweimal über den Atlantik ⇒ *langsam*



## Typische Probleme bei der Implementation

- Router upgraden → Flash, RAM, SW-Verfügbarkeit
- Hosts upgraden → Downtime
- Appliances (Print-Server, Webcams, ...)
- Software upgraden/portieren/patchen (z.B. qmail, squid)
- Doppelter Pflegeaufwand für IP- und Firewall-Konfiguration
- Tippfehler führen zu “verzögerten” Problemen (DNS z.B.)
- Seiten-Effekte von IPv6 autoconfig (z.B. Sniffing, Test-Router)
- Schulungsaufwand

## Work in Progress

- komplette DNS-Chain über v4 & v6-Transport (EDS0, truncating, balkanization)
  - v6-Transport zu den Root-Servern selbst
  - IPv6-Glue & v6-Transport für TLD-Server
  - Abläufe & Technik in den TLD-Registries
- IPv6-Address-Allokations-Policy bei den RIRs
  - Formulierungen z.T. unklar (/48s vs. /64s, 200 user)
  - Bestimmte Fälle gar nicht abgedeckt (site-local/PI)
  - Sub-Allocations undefiniert

## Offene Fragen: Multihoming

- Initiale Idee: Kundennetze nutzen /48s von mehreren ISPs
  - Source-IPs werden in Abhängigkeit von Ziel-IPs gewählt
  - Problem: Mechanismen dafür undefiniert
  - Problem: Routing-Policy verschoben auf Endgerät
  - Problem: Ausfall einer Leitung → TCP-Sessions sterben
- Stand heute: Kunden-Prefixe “klassisch” via BGP announced
  - pragmatische Lösung, Widerstand von “Theoretikern”
- IETF: Ziele für eine Multihoming-Lösung in RFC 3582
- IETF multi6: Trennung von “locator” & “identifier”
  - derzeit zwei Design-Teams aktiv
  - möglicherweise massive Änderungen an IP-Stacks nötig

## Offene Fragen: Firewalling?

- Verfügbarkeit von “Firewalling”
  - BSDs: ip6fw, Linux: netfilter (ohne conntrack?)
  - Cisco, Juniper: einfaches Packet-Filtering
  - Cisco PIX, Checkpoint FW, Netscreen: nur Beta-Versionen
- Paradigma-Wechsel durch IPSEC
  - verschlüsselte Ende-Ende-Kommunikation erwünscht
  - $\Rightarrow$  durch klassische Firewalls keine Dienstfilterung möglich
- Probleme auch durch unbekannte Extension Header möglich
- $\Rightarrow$  neue Ansätze nötig
  - Distributed Firewall?

## Links zu IPv6

- <http://hs247.com> - aktuelle IPv6 news
- <http://www.6bone.net/> - 6Bone, viel Doku
- <http://www.6bone.net/misc/case-for-ipv6.html>
- <http://www.ripe.net/rs/ipv6/ipv6allocs.html>
- <http://www.sixxs.net/tools/grh/> - ghost route hunter
- <http://www.space.net/~gert/RIPE/>
- <http://www.kame.net/> - v6 apps, BSD stacks
- <http://www.bieringer.de/linux/IPv6/> - Linux + IPv6
- <http://www.cisco.com/go/ipv6> - Cisco IPv6 Doku

**Fragen?**

`gert@net.in.tum.de`