

IP Version 6

Das neue Internetprotokoll

Vorlesung

Kommunikationssysteme: Aktuelle Themen aus Technik und Anwendungen

Dezember 2011

Guido Wessendorf
Zentrum für Informationsverarbeitung
Westfälische Wilhelms-Universität Münster
wessend@uni-muenster.de

wissen.leben
WWU Münster

Themen

- Wissenschafts-Netz Münster
- Warum ein neues Internet-Protokoll?
- Features von IPv6
- Transition von IPv4 nach IPv6
 - Rückblick: JOIN-Projekt
- Aufwand
- IPv6 an der WWU Münster
- Host-Beispiele

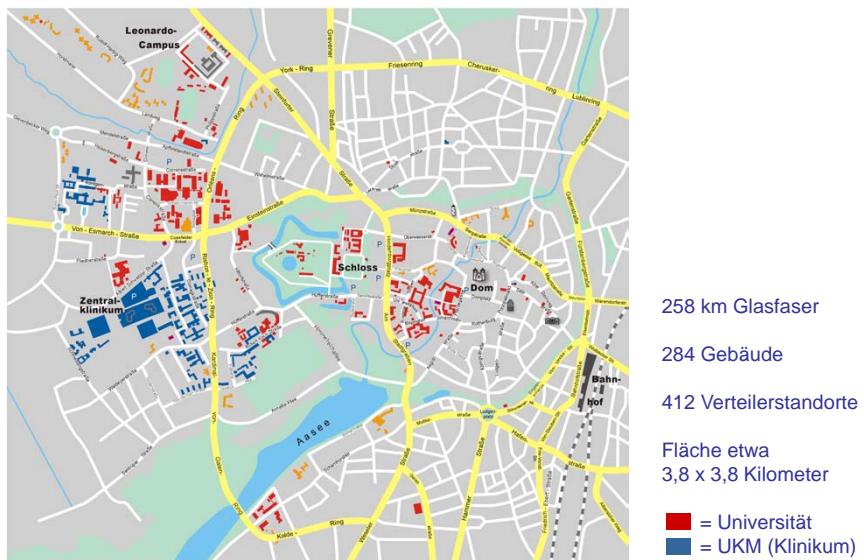
WissenschaftsNetz Münster (WNM)



Großes Stadt-Netzwerk

- gemeinsames Netz:
 - Westfälisches Wilhelms-Universität (WWU)
 - Universitäts-Klinikum Münster (UKM)
- geplant, aufgebaut und betrieben vom ZIV
- Zahlen:
 - 37.200 Studenten, 12.000 Mitarbeiter
 - 32.470 registrierte Endsysteme
 - 284 erschlossene Gebäude
 - 54.539 LAN-Dosen
 - 20 Core-Router (zurzeit Cisco Catalyst 6509)
 - 1.700 Switche
 - 1.353 Wlan Access-Points
 - 19.019 analoge/digitale Telefone

Großes Stadt-Netzwerk (2)

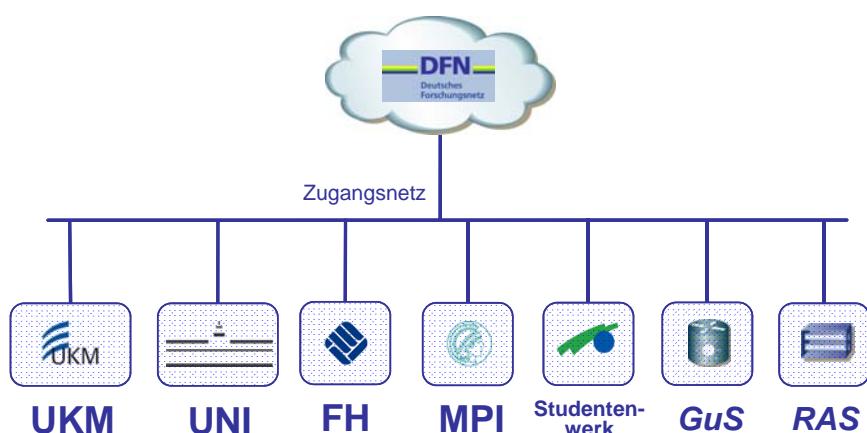


Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

5

ZENTRUM FÜR
INFORMATION
VERARBEITUNG

WissenschaftsNetz Münster (WNM)

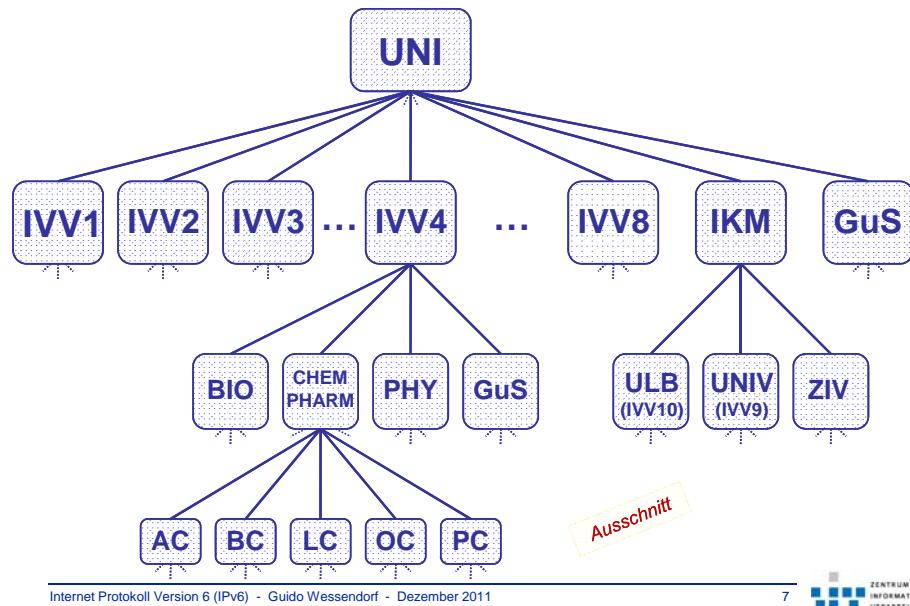


Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

6

ZENTRUM FÜR
INFORMATION
VERARBEITUNG

IV-Organisation an der Uni



Warum ein neues Internet-Protokoll?

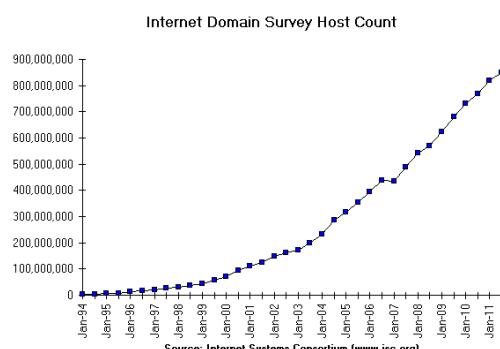
Famous Last Words

- „I think there is a world market for maybe five computers.“
– *Thomas Watson, chairman of IBM, 1943*
- „640K ought to be enough for anybody.“
– *Bill Gates, 1981*
- „32 bits should be enough address space for Internet.“
– *Vint Cerf, 1977*

Internet Wachstum

IPv4 wurde Anfang der 80'er Jahre entwickelt

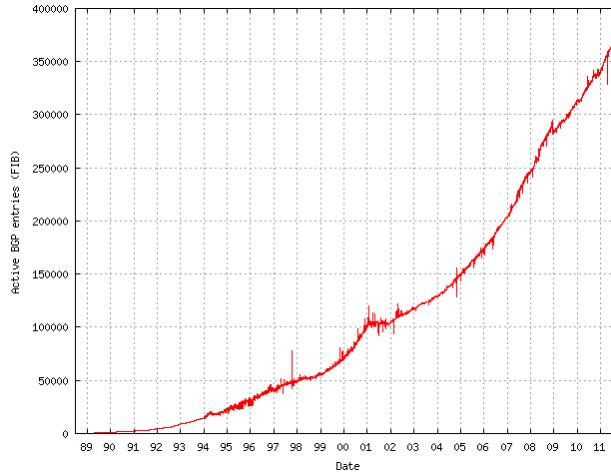
- damals heutige Internet-Größe und heutiges Internet-Wachstum nicht vorhersehbar:
 - im Januar 2011: ≈ 818 Millionen Hosts (DNS)
 - Verdopplung zurzeit etwa alle 3-4 Jahre!



Quelle: <https://www.isc.org/solutions/survey>

Explosion der Routing-Tabellen-Größe

- Router haben zurzeit 385.865 Inter-Domain-Routing-Einträge (7.12.2011)
 - großes administratives und Performance-Problem



Quelle: <http://www.cidr-report.org/>

IPv4 Adress-Dilemma

- obwohl mit 32-Bit IPv4-Adressen theoretisch etwa 4 Milliarden Systeme adressiert werden können, ist der IPv4-Adressraum bald erschöpft
 - ALE-WG, IETF-Meeting Juli 1994 Toronto:
 - Gesamt-IPv4-Adressraum zwischen 2005-2011 erschöpft
- Heise-News 18.01.2010: "RFC gibt reservierte IPv4-Adressen frei" (siehe <http://www.heise.de/netze/meldung/RFC-gibt-reservierte-IPv4-Adressen-frei-906971.html>)
- Meldung NRO 18.10.2010: „Remaining IPv4 Address Space Drops Below 5%“ (siehe <http://www.nro.net/media/remaining-ipv4-address-below-5.html>)
- Heise-News 8.11.2010: "Chinas Netzbetreiber drängen auf IPv6"
 - [...] Ende 2011 werden der China Telecom nach Angaben von Zhao Hui-Ling, Vizepräsidentin des Beijing Research Institute (BRI) von China Telecom, rund 20 Millionen IP-Adressen fehlen" [...](siehe <http://www.heise.de/newssticker/meldung/Chinas-Netzbetreiber-draengen-auf-IPv6-1132493.html>)
- Heise-News 24.3.2011: "Nortel verkauft eigene IPv4-Adressen an Microsoft"
 - 666.624 IPv4 Adressen für 11,25 US-\$ pro Adresse(siehe <http://www.heise.de/newssticker/meldung/Nortel-verkauft-eigene-IPv4-Adressen-an-Microsoft-1214670.html>)
- Intec NetCore, Inc.: IPv4 Exhaustion Counter (siehe http://inetcore.com/project/ipv4ec/index_de.html)

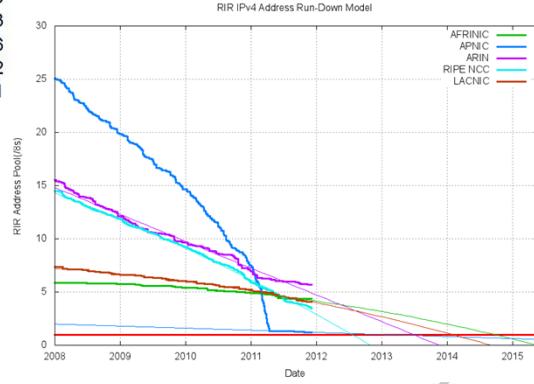


IPv4 Adress-Dilemma (2)

- Adressraum der IANA **reichte(!)** bis 3. Februar 2011
- Adressraum der regionalen Internet-Registries (RIR):

RIR	Projected Exhaustion Date
APNIC:	19-Apr-2011
RIPENCC:	18-Jul-2012
ARIN:	26-Jun-2013
LACNIC:	01-Feb-2014
AFRINIC:	22-Sep-2014

RIR	Remaining Addresses in RIR Pool (/8s)
	1.2009
	3.4513
	5.7016
	4.0902
	4.3621



Quelle: <http://www.potaroo.net/tools/ipv4/>

Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

13

ZENTRUM FÜR
INFORMATION
VERARBEITUNG

Fazit

- Skalierungs-Probleme von IPv4 lösen
- neue Technologien/Features besser unterstützen
 - Stichworte: Security, Mobility, Plug&Play, Multimedia-Anwendungen, Quality-of-Service
- ⇒ neues Internet-Protokoll notwendig!

Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

14

ZENTRUM FÜR
INFORMATION
VERARBEITUNG

Der Anfang ...

- Ende 1991 startete die IETF ihre Bemühungen um einen IPv4-Nachfolger:
IP Next Generation (IPng)
- im Juli 1992 wurden mehrere IPng-Proposals in Auftrag gegeben und Working-Groups eingerichtet
- im Juli 1994 auf dem IETF-Meeting in Toronto wurde dann **IPv6** als Nachfolger von IPv4 ausgewählt
(IPv6 = Internet Protocol Version 6)

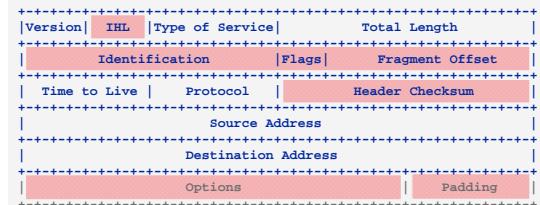
Features von IPv6

IP Version 6 – die wichtigsten Features

- erweiterte Adressierungs- und Routing-Möglichkeiten
- vereinfachtes Header-Format
- verbesserte Unterstützung von Optionen und Erweiterungen
- Flow-Label (effiziente Unterstützung von QoS)
- verbesserte Unterstützung von Security, Autoconfiguration (“Plug&Play”) und Mobility
- schrittweise ins Internet einföhrbar!

IPv4 Header

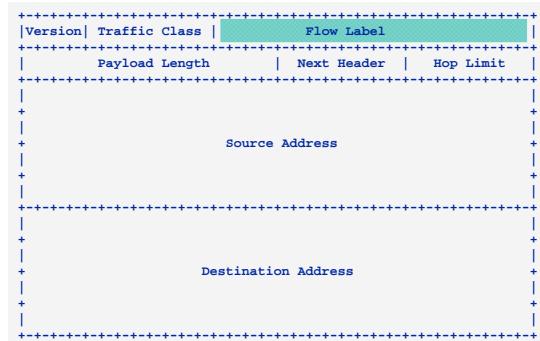
■ = aus Basisheader entfernt



IPv6 Header

■ = neues Feld

fast alle anderen Header-Felder
umbenannt und ggf. inhaltlich
angepasst ...



Extension Header – ein neues Design-Konzept

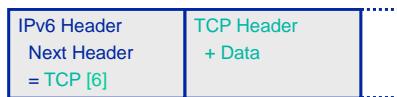
- Optionen/Zusätze werden *bei Bedarf* als verkettete Liste dem Basis-Header angehängt
- Basis-IPv6-Header bleibt mit 40-Oktett minimal und von konstanter Länge
- Verarbeitung nur beim Ziel-Node (Destination-Address) (Ausnahme: „Hop-by-Hop“- und Routing-Optionen)
- eliminiert IPv4's 40-Oktett-Limit für Optionen
- zukunftssicher, da leicht weitere Extension-Header definierbar

Extension Header

- Hop-by-Hop Options [0]
 - u. A. Handling von „Jumbo Payload“
 - Routing (Typ-0) [43]
 - Fragment [44]
 - Authentication (AH) [51]
 - Encapsulating Security Payload (ESP) [50]
 - Destinations Options [60]
- ✓ eine „vollständige“ IPv6-Implementierung liegt nur dann vor, wenn mindestens *alle* obigen Extension-Header implementiert sind!

Extension Header Liste – Beispiele

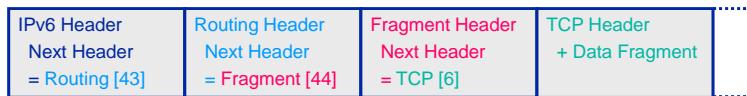
- Minimal



- Source-Routing



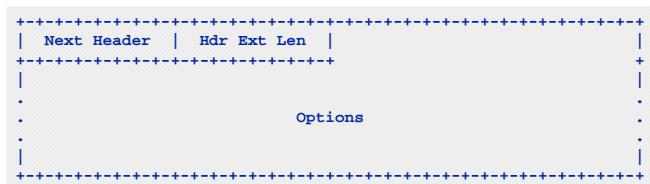
- Source-Routing + Fragmentierung



Extension Header im Detail - Beispiele

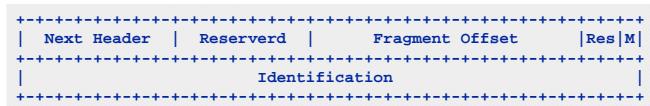
- Destination Options (Type=60)

- transportiert optionale Informationen, die ausschließlich zur Bearbeitung beim Ziel bestimmt sind



- Fragment (Type=44)

- falls Pakete aufgrund der Größe nicht übertragen werden können (Path MTU), werden sie in mehrere kleinere Pakete aufgeteilt (fragmentiert)
- im Gegensatz zu IPv4 wird Fragmentierung *nur* vom Absender durchgeführt
- M-Flag: 1 = mehr Fragmente, 0 = letztes Fragment



IPv6 Adressen

Die neuen IPv6-Adressen

- IPv4
 - **128.176.191.6**
 - dezimale Notation
 - 32 Bit lang
 - » entspricht etwa 4 Milliarden möglichen Adressen
- IPv6
 - **2001:638:500:101:250:56ff:fead:7f9**
 - hexadezimale Notation
 - 128 Bit lang
 - » 2^{128} Adressen
 - etwa 340 Sextillionen Adressen
 - etwa 667 Billiarden Adressen/mm² Erdoberfläche

IPv6 Adressen – Repräsentation

- „Normal“
 - 8 Blöcke á 16 Bit in hexadezimaler Notation getrennt durch „:“
 - führende Nullen können weggelassen werden
$$\begin{array}{l}
 \text{3ffe:0400:0010:0100:0a00:20ff:felb:d640} \\
 \Leftrightarrow \text{3ffe:400:10:100:a00:20ff:felb:d640}
 \end{array}$$
 - Komprimiert
 - aufeinanderfolgende Nullen-Gruppen können durch „::“ zusammengefasst werden
$$\begin{array}{l}
 \text{ff02:0:0:0:0:0:0:2} \qquad \qquad \qquad \text{0:0:0:0:0:0:0:1} \\
 \Leftrightarrow \text{ff02::2} \qquad \qquad \qquad \Leftrightarrow \text{::1}
 \end{array}$$
 - IPv4-Kompatibel (Spezialsyntax für enthaltene IPv4-Adressen)

$$\begin{array}{l}
 \text{0:0:0:0:0:128.176.191.66} \\
 \Leftrightarrow \text{::128.176.191.66}
 \end{array}$$
 - Adress-Präfix (Subnetz-Angabe)
 - CIDR Notation (wird auch in Host-/Router-Konfigurationen verwendet)
$$\begin{array}{l}
 \text{3ffe:0400:0010:0100:0a00:20ff:felb:d640/48} \\
 \Leftrightarrow \text{3ffe:400:10::/48}
 \end{array}$$

IPv6 Adresstypen

- Unicast (one-to-one)
 - link local
 - unified local (ULA, ephemeral site local)
 - global
 - (IPv4-) compatible
 - Anycast (one-to-nearest)
 - Multicast (one-to-many)

Adresstyp-Präfixe

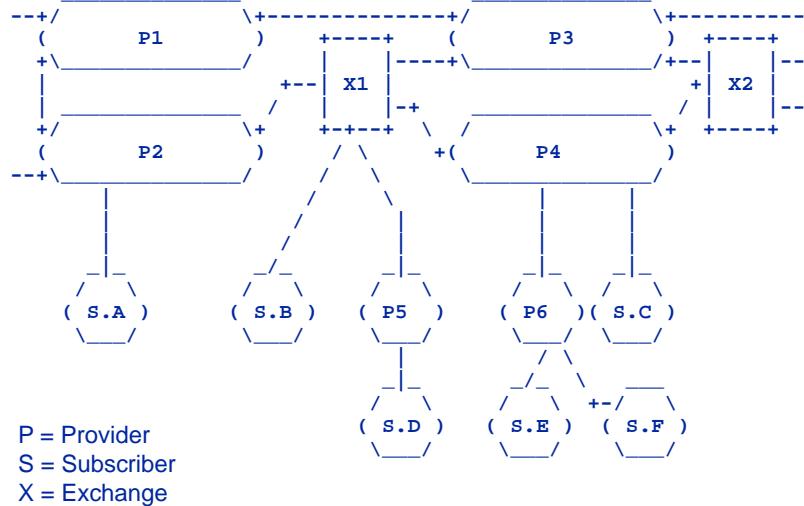
RFC 4291

Adress-Typ	Bit-Präfix	Präfix-Notation
• Unspecified	0...0	<code>::/128</code>
• Loopback	0...01	<code>::1/128</code>
• Multicast	<code>11111111</code>	<code>ff00::/8</code>
• Link Local	<code>1111111010</code>	<code>fe80::/10</code>
• Unique Local (ULA)	<code>1111110</code>	<code>fc00::/7</code>
• Global Unicast	(alles andere)	
– IANA delegiert zurzeit 2000::/3 (Bit-Präfix=001), der Rest ist reserviert!		

Hierarchisches Adresspräfix-Routing

- Zusammenfassen mehrerer Adressen unter einem Präfix
 - nur das aggregierte Präfix darf für das Routing angegeben werden
 - dadurch bleibt globale Routingtabelle kleiner
- Adressen sind topologisch signifikant
 - Providerabhängig!
- Konsequenz: Topologieänderung erfordert Umadressierung
 - IPv6 ermöglicht leichteres Renumbering
- Adressen dienen zur eindeutigen Identifizierung und zur topologischen Lokalisierung

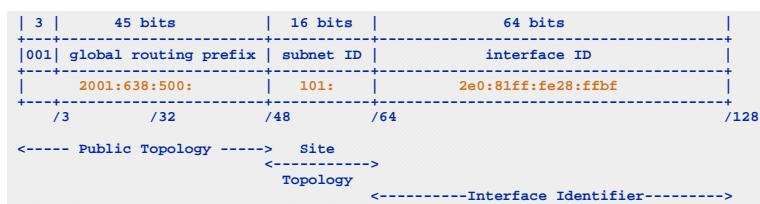
Beispiel einer Public-Topology



Globale Unicast Adressen

RFC 3587

- RFC legt allgemeine Struktur fest;
hier eine in der Praxis etablierte „Real-World“ Aufteilung:



- feste vorgegebene Trenngrenze bei 64 Bit (/64):
 - obere Adress-Hälfte: Routing
 - untere Adress-Hälfte: Interface-Adressierung
- zurzeit praktizierte Default-Grenze für Provider sind 32 Bit (/32)
- empfohlene Grenze bei 48 Bit (/48) für Endeinrichtungen (Subscriber/Kunden)

Beispiele von Adressaggregierungen

WWU-DFN-Präfix

- RIPE NCC: 2001:600::/23
- DFN: 2001:638::/32
- WWU Münster: 2001:638:500::/48
- JOINsrv Subnetz: 2001:638:500:101::/64
- News-Server: 2001:638:500:101:2e0:81ff:fe28:ffbf

WWU-RIPE-Präfix

- RIPE NCC: 2001:4c00::/23
- WWU Münster 2001:4cf0::/32

Interface IDs

- 64-Bit lang (untere Hälfte der IPv6-Adresse)
- identifizieren ein Interface auf einem Link eindeutig
- werden beispielsweise wie folgt gebildet
 - IEEE EUI-64 Format
 - Privacy Extensions (RFC 4941)
 - manuell konfiguriert
 - DHCPv6

Generieren einer global gültigen EUI-64 Interface-ID aus Ethernet-Adresse

RFC 4291

IEEE 802 MAC-Identifier

0	1 1	3 3	4
0	5 6	1 2	7
+	-----	-----	-----
cccccc0gccccccc ccccccccc mmmmmmmm			
+	-----	-----	-----

0 = universal / 1 = local
g = individual/group
c = company
m = manufacturer

invertieren

einfügen

0	1 1	3 3	4 4	6
0	5 6	1 2	7 8	3
+	-----	-----	-----	-----
cccccc1gccccccc ccccccccc11111110 11111110mmmmmm				
+	-----	-----	-----	-----

IEEE EUI-64 Identifier

ff fe

- Beispiel: MAC = 00:c0:9f:52:81:fd
⇒ EUI-64 = 2c0:9fff:fe52:81fd

Privacy Extensions

RFC 4941

- Sicherheits-Problem mit EUI-64-Adressen
 - Interface-MAC in Absender-IP enthalten
 - » Rückschlüsse auf Hardware/Betriebssystem
 - » User-Tracking (z.B. Wechsel Netzwerk oder Provider, etc.)
- Lösung: Interface-ID zufällig generieren und regelmäßig wechseln
- typisch: Source-IP täglich wechseln, bleibt aber bzgl. Empfang 7 Tage gültig (konfigurierbar)
- implementiert in nahezu allen Betriebssystemen
 - default aktiviert: Windows-XP, -Vista, -7 und iOS 4.3

Privacy Extensions (2)

- Beispiel: Windows 7

```
C:\> netsh interface ipv6 show privacy
Der aktive Status wird abgefragt...
Parameter für temporäre Adressen
-----
Temporäre Adresse verwenden : enabled
Versuch, doppelte Adr. zu entdecken : 5
Maximale Gültigkeitsdauer : 7d
Maximale bevorzugte Gültigkeitsdauer: 1d
Regenerationszeit : 5s
Maximale Verzögerungszeit : 10m
Verzögerungszeit : 0s

C:\> ipconfig
Windows-IP-Konfiguration
Ethernet-Adapter LAN-Verbindung:
  Verbindungsspezifisches DNS-Suffix: example.net
    IPv6-Adresse . . . . . : 2001:db8::1:9db9:d778:2de1:60c7
    Temporäre IPv6-Adresse. . . . . : 2001:db8::1:f4a5:3977:8b75:b95a
    Verbindungslokale IPv6-Adresse . . : fe80::9db9:d778:2de1:60c7%13
    IPv4-Adresse . . . . . : 192.168.1.36
    Subnetzmaske . . . . . : 255.255.255.0
    Standardgateway . . . . . : fe80::2cf:ddff:fea:b443%13
                                192.168.1.1
```

siehe Heise-Artikel: [IPv6: Privacy Extensions einschalten](#)

Link Local Adressen

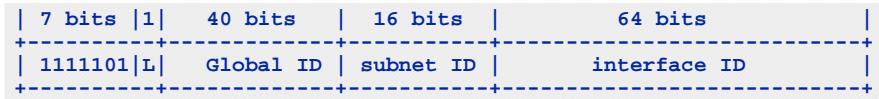
RFC 4291



- Präfix ist `fe80::/10`
- Adressen nur auf einzelnen Link gültig
- jedes Interface bekommt automatisch eine Link Local Adresse
- Verwendung z.B. bei Adress-Autokonfiguration, Neighbor-Discovery oder wenn kein Router vorhanden ist
- Router dürfen IPv6-Pakete mit Link Local Adressen nicht weiterleiten

Unique Local Unicast Addresses (ULA)

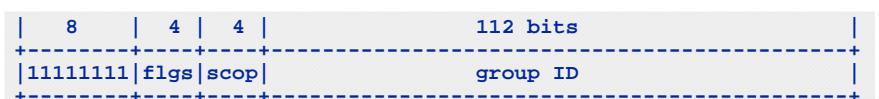
RFC 4193



- „private“ IPv6-Adressen (anfangs auch SLA – Site Local Addr. genannt)
- Präfix ist `fc00::/7`
- L-Bit:
 - zentral organisierte Adresse: `L=0` -> `fc00::/8`
 - „ULA-Central“
 - lokal bestimmte Adresse: `L=1` -> `fd00::/8`
- Global ID:
 - zufällig erzeugter Präfix, das nur innerhalb der Site genutzt wird
- ULA-Central Adressen werden nicht global geroutet, sind aber global gültig und eindeutig

Multicast Adressen

RFC 4291



- Präfix ist `ff00::/8`
- Flags: weitere Attribute der Adresse (u.A. permanent/local gültig)
- Scope
 - 1 interface-local
 - 2 link-local
 - 4 admin-local
 - 5 site-local
 - 8 organisation-local
 - E global
- Beispiele
 - `ff02::1` alle Nodes auf dem Link
 - `ff02::2` alle Router auf dem Link
 - `ff05::1:3` alle DHCPv6-Server der Site
 - `ff0x::1:101` alle NTP-Server (x=Scope)
 - `ff02::1:ffxx:xxxx` Solicited Multicast (x=letzten 3 Oktett der Adresse)

Plug & Play

Eingebaute Autoconfiguration Features

Neighbor Discovery (ND)

- ND integraler Bestandteil einer IPv6-Implementierung
- Basis-Funktionen:
 - Stateless Adress Autoconfiguration (SLAAC)
 - ermitteln von Nachbar-Routern
 - ermitteln der Link-Layer-Adresse (MAC) eines Nachbarn (Neighbor Solicitation)
 - verwalten der Nachbarliste (Neighbor Cache)
 - prüfen der Erreichbarkeit von Korrespondenz-Nachbarn (Neighbor Unreachable Detection, NUD)
 - Redirect
 - Duplicate Address Detection (DAD)
- ND basiert auf dem ICMPv6-Protokoll

Stateless Address Autoconfiguration (SLAAC)

- Plug & Play: keine zusätzlichen Server und wenig oder keine administrativen Eingriffe notwendig
- für jedes Host-Interface:
 - bilden einer Link-Local-Adresse
 - senden einer Router Solicitation (RS, ICMP-Type 133) zur All-Router-Multicast-Adresse (ff02::2)
 - Router sendet(n) Router Advertisement(s) (RA, ICMP-Type 134) mit Präfix-Information(en)
 - Router-Präfix(e)+Interface-ID bilden die zusätzliche(n) Interface-Adresse(n)
 - jede so gewonnene Adresse wird auf Eindeutigkeit auf dem Link überprüft (Duplicate Address Detection, DAD)
- Hinzufügen neuer Subnetz-Präfixe im Router im Zusammenspiel mit Preferred-/Valid-Lifetime-Zählern der Präfixe bewirkt automatisches Renumbering im Subnetz ...

Neighbor Solicitation (NS)

- Funktionalität ähnlich ARP bei IPv4 (Internet Standard 37)
- Ablauf (Host A benötigt MAC-Adresse von Host B):
 - senden eines NS-Paketes (ICMP-Type 135)
 - » Absender = Host A
 - » Ziel = „Solicited-Node Multicast“-Adresse von B:
ff02::1:ff_{xx:xxxx}
wobei _{xx} die unteren 24-Bit der IPv6-Adresse(n) von B
 - Antwort von B (ICMP-Type 136)
 - » Absender = Host B
 - » Ziel = Host A
 - » Data = MAC-Adresse von B

Der neue Flow Label

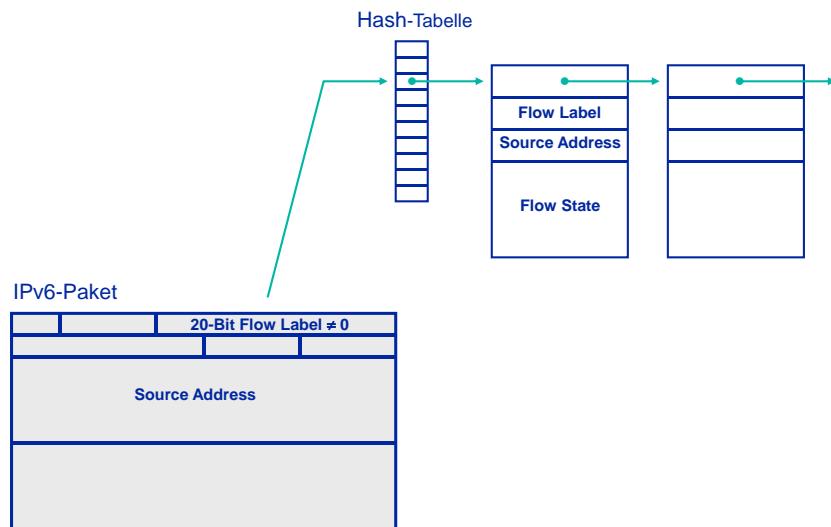
Flows in IPv6

Allgemein ist ein Flow ein “zusammengehöriger” Datenstrom.

Für QoS-Steuerung bei IPv6-Kommunikation macht folgende Beschreibung Sinn:

Ein Flow ist eine Folge von Paketen von einem bestimmten Absender zu einem bestimmten (Unicast- oder Multicast-) Ziel, für die der Absender eine spezielle Behandlung bei der Übertragung durch Router wünscht, wie z. B. Nicht-Default oder Real-Time-Services für Audio/Video-Übertragungen.

Flow-State-Lookup im Router



Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

45

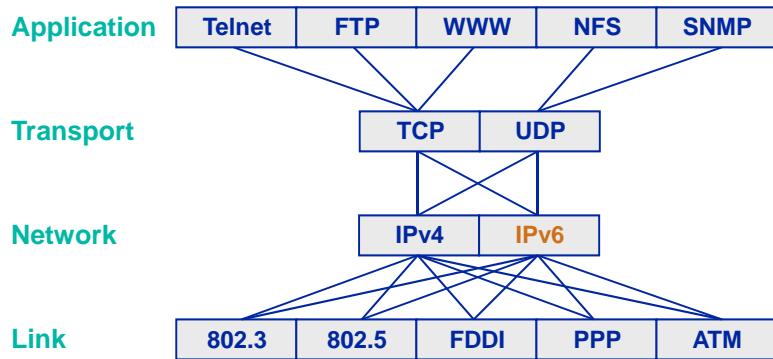
ZENTRUM FÜR
INFORMATION
VERARBEITUNG

Transition von IPv4 nach IPv6

ZENTRUM FÜR
INFORMATION
VERARBEITUNG

Transition IPv4 > IPv6 (1)

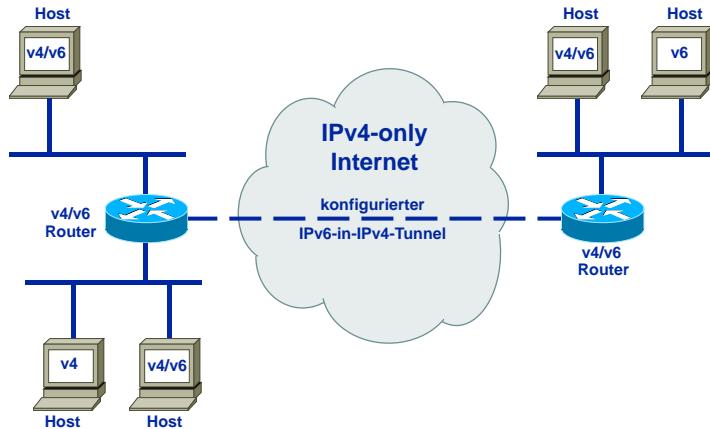
- Grundkonzept: *Dual-IP-Stack*
 - IPv4-Hosts und -Router werden nach und nach um IPv6-Stack ergänzt
 - Software/Applikationen müssen ggf. angepasst werden!



Transition IPv4 > IPv6 (2)

- Anfangs: Überbrücken von (noch) nicht IPv6-fähigen Topologien durch Tunnel-Technik:
 - Tunnel durch IPv6-in-IPv4-Encapsulation
 - Tunnel-Methoden:
 - » konfigurierte Tunnel
 - die IPv4-Tunnelendpunkte werden manuell konfiguriert; die Routingtabelle gibt vor, welche IPv6-Pakete über welchen Tunnel transportiert werden
 - » automatische Tunnel
 - diverse weitere Methoden (s.u.)

Transition IPv4 > IPv6 (3)



Transition IPv4 > IPv6 (4)

- sobald Router um IPv6-Stack ergänzt sind, können IPv6-in-IPv4-Tunnel wieder entfernt werden
 - Ziel: native IPv6-Konnektivität!
- IPv6 wird sich neben IPv4 im Internet etablieren
- IPv4 wird noch viele Jahre im Internet geroutet werden
- IPv4-in-IPv6-Tunnel um später IPv4-Inseln über IPv6-Topologien zu verbinden

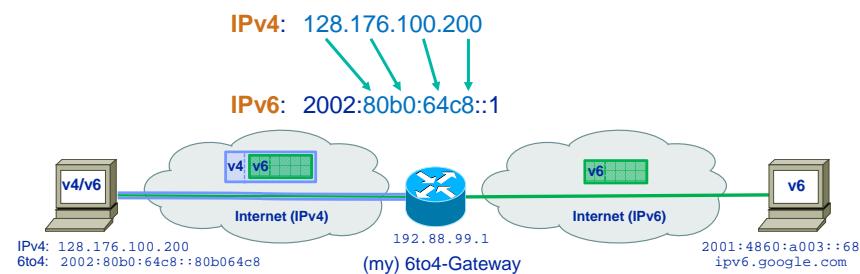
Transition-Mechanismen (Zusammenfassung)

- Dual-IP-Stack
- Tunnel-Techniken
 - konfigurierte Tunnel
 - 6to4
 - » 6RD
 - Teredo
 - ISATAP
 - Tunnelbroker
- Translatoren
 - IP-Level (NAT-PT, SIIT)
 - TCP-Level (TCP-Relay, Socks)
 - Application-Level (ALG, Bump in the API, Proxy)
 - DSTM

6to4

RFC 3056

- ist ein automatischer Tunnelmechanismus, um auch ohne native IPv6 Konnektivität IPv6-Pakete transportieren zu können
- globale IPv4-Adresse nötig; NAT nur, wenn Gateway [proto41-nat] unterstützt
- IPv4 Protokolltyp 41
- Funktionsweise:
 - 6to4-IPv6-Präfix: 2002::/16
 - 6to4-IPv4-Gateway: 192.88.99.1 (Anycast-Adresse)
 - Adress-Mapping einer IPv4-Adresse zu einer 6to4-IPv6-Adresse, Beispiel:



ISATAP (IntraSite Automatic Tunneling Addressing Protocol)

RFC 5214

- einfachste Methode um ganze Netze v6-fähig zu machen
- ISATAP definiert eine Methode, um Link-Lokale IPv6-Adressen aus IPv4-Adressen zu generieren und darauf basierend Neighbor-Discovery zu machen
 - Link-Lokale Interface-Adressen werden gebildet aus
 - » `fe80::5ef:e<32-Bit IPv4-Adresse>`
- welcher ISATAP-Router?
 - Konfiguration einer PRL (Potential Router List) oder z.B. `isatap.uni-muenster.de`
 - Vorsicht: Wenn definiert und Tunnel funktionsfähig, dann ggf. bevorzugte Verbindung!

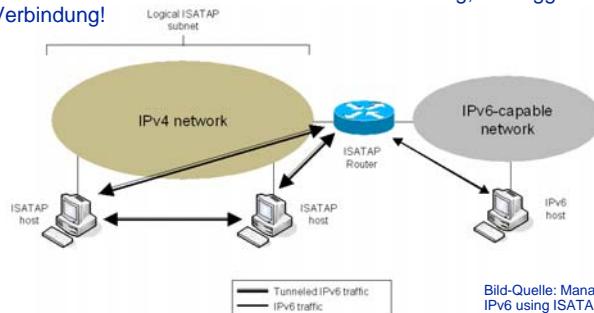


Bild-Quelle: Manageable Transition to IPv6 using ISATAP (Microsoft/Cisco)

Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

53

ZENTRUM FÜR
INFORMATIONSVERARBEITUNG

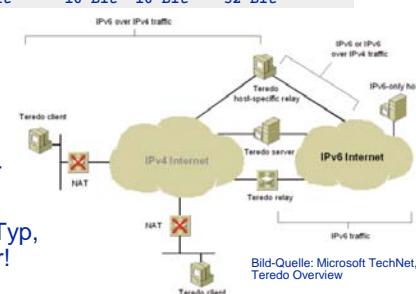
Teredo

Teredo = Schiffsbohrwurm (Muschel ;-)

RFC 4380

- Tunneltechnik, die auch bei NAT (Network Address Translation) funktioniert
 - typisches Heim-Szenario: nur DSL-Router hat öffentliche IP-Adresse
- IPv6-Pakete werden in IPv4-UDP-Pakete getunnelt mithilfe von
 - Teredo-Servern (Adress-Registrierung, Vermittlung) und
 - Teredo-Relays (falls Ziel IPv6-only erreichbar ist)
 - Teredo-Server verwendet UDP-Port 3544
 - Teredo-Service-Präfix: `2001:0000::/32`
- Teredo-Adressen:

2001:0000	Server IPv4	Flags	Port	Client IPv4
32 Bit	32 Bit	16 Bit	16 Bit	32 Bit
- Windows: sobald eine native IPv6-, 6to4- oder ISATAP-Konnektivität zur Verfügung steht, arbeitet der Host *nicht* als Teredo-Client
- Windows: Default-IPv4-Teredo-Server-Adresse ist `teredo.ipv6.microsoft.com`
- Vorsicht: Verkehr nicht erkennbar am Typ, verhält sich wie normaler UDP-Verkehr!

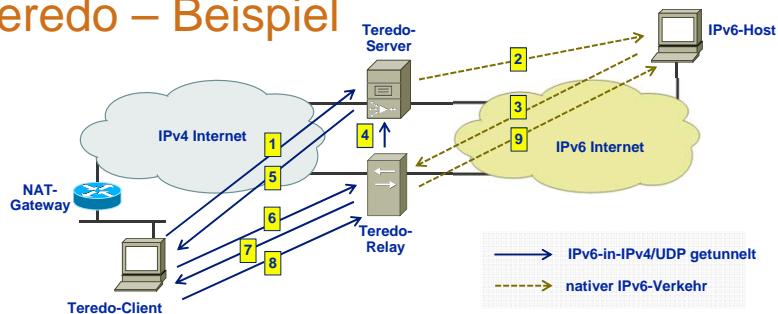


Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

54

ZENTRUM FÜR
INFORMATIONSVERARBEITUNG

Teredo – Beispiel



Warum kommt IPv6 so zögerlich?

Aufwand und Restriktionen

- Upgrade von Hard- und Software
- Schulung des Personals
- umfangreiche Planungen
 - Netzkonzepte
 - » Adressraumplanung (Präfixe)
 - » Adress-Vergabe (SLAAC, DHCPv6, EID)
 - » Routingprotokolle (BGP, IS-IS, OSPFv3, RIPng)
 - Sicherheit (Netze dürfen durch IPv6 nicht unsicherer werden als es für IPv4 als für nötig erachtet wurde!)
 - » leider oft mangelhafte IPv6-Fähigkeit spezieller Sicherheitsinstanzen (z.B. Firewalls, Intrusion Prevention Systeme, VPN)
- Durchführung von Tests
 - Labor-, Test- und Pilotbetrieb
- Netzmanagement (Datenbanken, Monitoring, Reporting, Accounting)
- Schulung der Kunden
- „Business Case“ ???

Probleme der Content-Provider

- viele große Internet-Sites (Content-Provider) führen nur zögerlich IPv6 im Dualstack-Betrieb ein. Es besteht die Befürchtung, Kunden durch Probleme mit IPv6 zu verlieren
 - angeblich seien ~0,03% der Nutzer möglicherweise betroffen (und dann aber auch permanent)
 - die meisten aktuellen Betriebssysteme versuchen IPv6 zuerst und fallen erst im Problemfall auf IPv4 zurück
 - schnelle lokale Fehler durchaus normal, kein Problem...
 - langsame Fehler (Timeouts) problematisch
 - » Windows: 20 Sekunden
 - » OS X: 4 oder 75 Sekunden
 - » Linux: 3 Minuten oder dauerhaft
- (diese Zeiten gelten für jede Verbindung, und eine Webseite hat viele!)

Mögliche Ursachen

- Implementierungsfehler, u. A.
 - Heim-Router senden RA trotz fehlender Verbindung
 - Hosts ignorieren ICMPv6-Unreachable Meldungen
 - Heim-Router DNS-Proxy unterstützt lange AAAA-Antworten nicht
- Fehlkonfiguration, u. A.
 - 6to4 Tunnel-Mode hinter NAT-Gateway
- Kunde: „Seite geht nicht!“
 - es werden selten lokale Probleme vermutet: andere (IPv4-)Seiten gehen ja ...

Feldtests

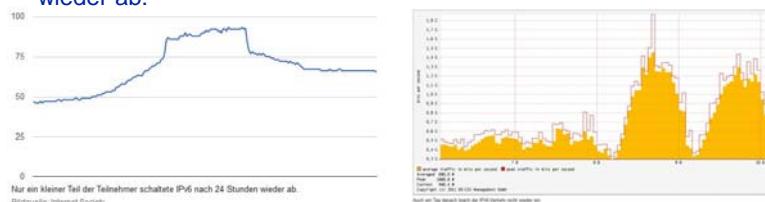
Damit das Risiko Kunden zu verlieren überprüft und eingeschätzt werden kann, werden gerne „Test-Tage“ durchgeführt. Beispiel:

- 16.9.2010: „IPv6-Tag“ bei www.heise.de
(rund 1 Millionen Besucher pro Tag, bei angenommen 0,1% mit Problemen wären das 1.000 abgehängte Leser)
 - „am 16.9.2010 behandeln wir IPv6 und IPv4 gleich“
 - umfangreiche Ankündigung und Feedback-Hotline
 - Ergebnis: nur 5 Kunden hatten Probleme, größtenteils durch Reboot, etc. leicht behebbar
 - seit 29.9.2010 dauerhaft Dualstack-Betrieb!

World IPv6 Day



- am 8.6.2011 fand ein weltweiter IPv6-Test statt
 - <http://isoc.org/wp/worldipv6day/>
- große Websides schalten auf Dualstack um, u. A.:
 - Facebook, Google, Yahoo, Akamai, Limelight Networks, Cisco, YouTube, Gmail, t-online.de, ...
- einfache Test-Seite: <http://www.test-ipv6.com>
- Heise-News 9.6.2011: „Viel Aufmerksamkeit und kaum Probleme“
- Heise-News 11.6.2011: „Unerwartete Nachwirkungen“
 - nur ein kleiner Teil der Teilnehmer schaltete IPv6 nach 24 Stunden wieder ab:



Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

61

ZENTRUM FÜR
INFORMATION
VERARBEITUNG

Rückblick: JOIN-Projekt

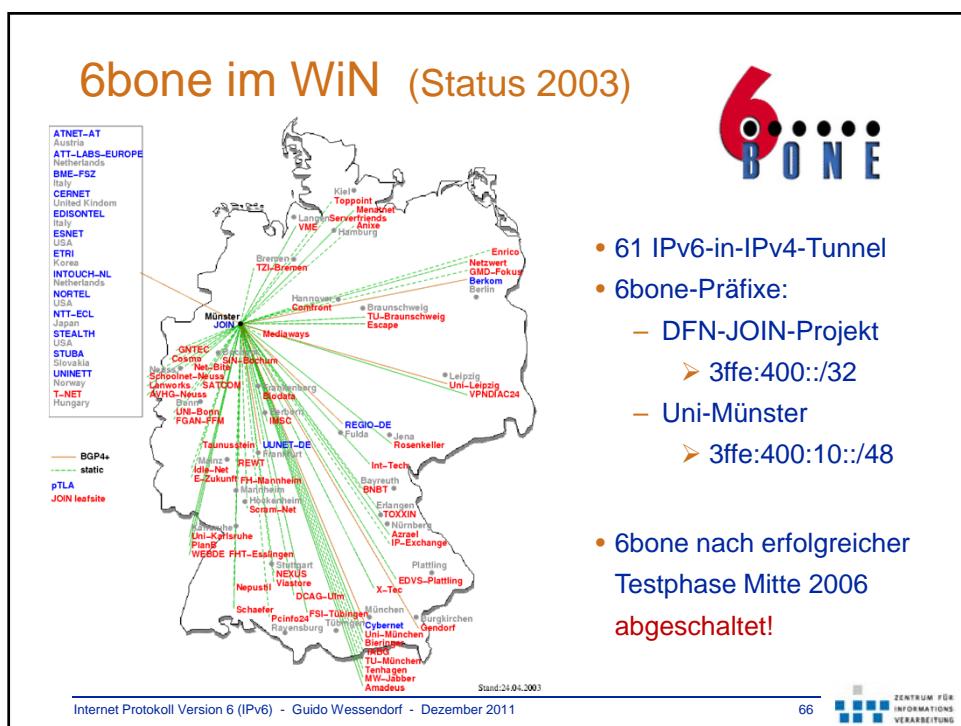
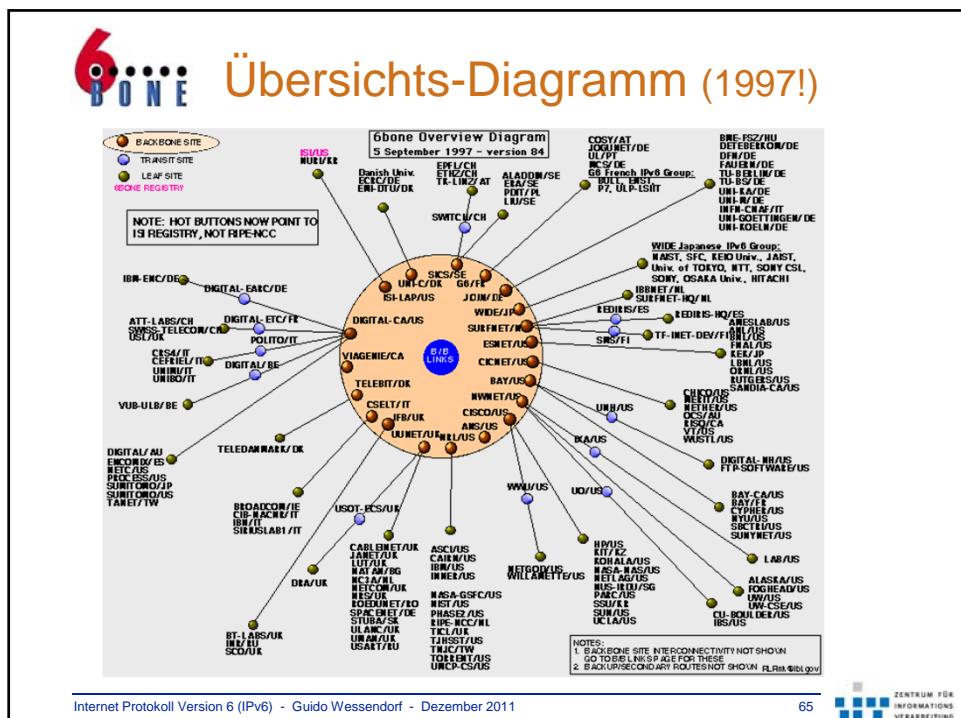
ZENTRUM FÜR
INFORMATION
VERARBEITUNG

JOIN-Projekt

- “IPv6 im WiN – JOIN-Referenzzentrum”
- ein Projekt des DFN-Vereins, durchgeführt von **Juli 1996 bis Mitte 2005** am Zentrum für Informationsverarbeitung (ZIV) der WWU Münster



- weltweites IPv6-Testnetzwerk
- realisiert die Transitionsstrategien
- am 15. Juli 1996 wurde der 6bone ins Leben gerufen
 - im Juni 2003 sind bereits 1.353 Teilnehmer (Organisationen/Netze) in 61 Ländern registriert!
- “JOIN”-Projekt war seit August 1996 deutsche Backbone-Site im 6bone



6WiN (Status 2004)

6WiN

Übersichtsbild des 6WiN



- 68 (58) vom DFN vergebene Präfixe.
- 35 (30) angeschlossene Einrichtungen
- 32 (26) über Tunnel/ 3 (4) nativ
- neu: 2 Multicast-Kunden: IABG Teleport, FhG Institutszentrum Birlighoven
- Status:
 - 6WiN wurde im Juni 2005 an DFN-NOC übergeben
 - seit 2007 IPv6 native im X-WiN

Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

67



IPv6 an der WWU Münster

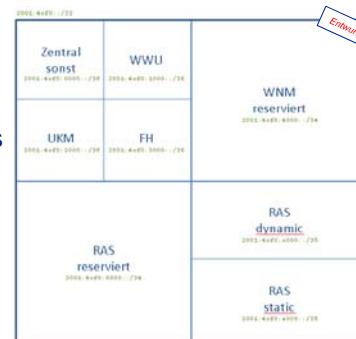
IPv6 an der WWU

- am ZIV IPv6 seit 1994 (wegen damaligem DFN JOIN-Projekt)
- bis dato „nur“ im Test-Betrieb, d.h. noch nicht (durchgängig) im produktivem Netzwerk-Core
- Sensibilisierung für IPv6
 - IPv6 kommt ...
 - möglichst frühzeitiges Know-How bei den IVVen/Admins!
 - » Weiterbildung
 - » Eigenerfahrung mit IPv6-Testbetrieb
 - auf „IPv6-Ready“ bei kommenden und existierenden Produkten achten
 - » Hardware (Drucker, ...)
 - » Software (Fremdsoftware und Eigenentwicklungen)
 - » berücksichtigen in Beschaffungen, Verträgen, Lizenzen auf IP-Adressbasis
 - Etablierung von IPv6 wird Ressourcen beanspruchen!
 - » bei allen, aber insb. an der WWU bei ZIV und IVVen und im UKM beim GB-IT
 - Nutzer müssen aufgeklärt werden
 - » Ankündigungen, Infos, Schulungs-Angebote
 - » Ziel: für Nutzer möglichst unbemerkt IPv6-Einführung
- umfangreiches ZIV-Projekt ab 2011



IPv6 an der WWU (2)

- Projekt 2011
 - IPv6 im Netzwerk-Core
 - » im UKM-Core optional
 - Vorbereitung Core-Network-Services
 - » DNS, DHCP, NTP
 - Vorbereitung NIC_online
 - erste Services nach außen
 - » ideal: Web-Serverpark
- Projekt 2012
 - sukzessive weitere Netzzonen (Servernetze, Arbeitsplätze)
 - » nur nach Rücksprache mit jeweiligen Verantwortlichen um IPv6-Routing erweitern
 - » unter Beachtung von Sicherheit und Funktionalität!
 - IPv6 ist optional



Host-Beispiele

Host-Beispiel: Windows XP

- ab Windows XP ist IPv6 integriert
 - allerdings noch nicht wie gewohnt via GUI, sondern nur via DOS-Eingabeaufforderung zu bedienen
- einige DOS-Kommandos:
 - `ipv6` (ohne Argument wird Hilfe angeboten)
 - `ipv6 install` (installiert IPv6-Stack)
 - `ipv6 uninstall` (entfernt IPv6-Stack)
 - `ipv6 if` (zeigt alle IPv6-Interface an)
 - `ipv6 rt` (zeigt IPv6-Routingtabelle an)
 - `ipv6 nc` (zeigt Neighbor-Cache an)
 - `netsh interface ipv6` (Net-Shell für IPv6 ...)
- Konnektivitäts-Tests:
 - `ping6`
 - `tracert6`
- u. A. Internet-Explorer 6 IPv6-fähig ...

Windows XP – ipv6

```
cmd
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Dokumente und Einstellungen\Administrator>ipv6 install
Installation wird durchgeführt...
Erfolgreich.

C:\Dokumente und Einstellungen\Administrator>
```



```
C:\Dokumente und Einstellungen\Administrator>ipv6
Syntax: ipv6 [-p] [-v] if [Schnittstellenindex]
        ipv6 [-p] ifcr u6v4 v4src vdst [Ind] [pmld]
        ipv6 [-p] ifcr hover4 v4src
        ipv6 [-p] ifc Schnittstellenindex [forwards] [-forwards] [advertisers] [-advertisers] [intu Anz. Bytes] [site Sitekennung] [preference P]
        ipv6 [-p] Schnittstellenindex v4dst
        ipv6 [-p] ifd Schnittstellenindex
        ipv6 [-p] add Schnittstellenindex/Adresse [life validlifetime[/preflifetime]] [anycast] [unicast]
        ipv6 nc [Schnittstellenindex] [Adresse]
        ipv6 ncf [Schnittstellenindex] [Adresse]
        ipv6 rc [Schnittstellenindex] [Adresse]
        ipv6 rcf [Schnittstellenindex] [Adresse]
        ipv6 hc
        ipv6 [-p] [-v] rt
        ipv6 [-p] rtu Präfix Schnittstellenindex[/Adresse] [life valid[/pref]] [preference P] [publish] [Alter] [spl Sitepräfixlänge]
        ipv6 spt
        ipv6 spu Präfix Schnittstellenindex [life L]
        ipv6 spv pp
        ipv6 [-p] spu [Parameter Wert] ... <eventuell -?>
        ipv6 renew [Schnittstellenindex]
        ipv6 [-p] ppt
        ipv6 [-p] ppn Präfix Reihenfolge P srclabel SL [dstlabel DL]
        ipv6 [-p] ppd Präfix
        ipv6 [-p] reset
        ipv6 install
        ipv6 uninstall
Einige Unterbefehle erfordern lokale Administratorrechte.
```

Windows XP – ipconfig

```
cmd
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Dokumente und Einstellungen\wessend>ipconfig

Windows-IP-Konfiguration

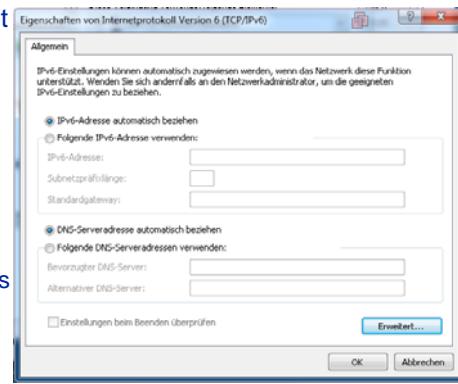
Ethernetaufgabe LAN-Verbindung:
  Verbindungsspezifisches DNS-Suffix: uni-muenster.de
  IP-Adresse . . . . . : 128.176.184.56
  Subnetzmaske . . . . . : 255.255.255.0
  IP-Adresse . . . . . : 2001:638:500:200:a587:3d16:5513:be3e
  IP-Adresse . . . . . : 2001:638:500:200:20c:29ff:fe90:a650
  IP-Adresse . . . . . : fe80::20c:29ff:fe90:a650%5
  Standardgateway . . . . . : 128.176.184.1
                                         fe80::208:e2ff:fe0e:2008%5

Tunnelaufgabe Teredo Tunneling Pseudo-Interface:
  Verbindungsspezifisches DNS-Suffix:
  IP-Adresse . . . . . : fe80::ffff:ffff:ffff:ffffd%4
  Standardgateway . . . . . :

Tunnelaufgabe Automatic Tunneling Pseudo-Interface:
  Verbindungsspezifisches DNS-Suffix: uni-muenster.de
  IP-Adresse . . . . . : fe80::5efe:128.176.184.56%2
  Standardgateway . . . . . :
```

Host-Beispiel: Windows Vista/7

- IPv6 standardmäßig eingeschaltet
- in GUI integriert
- weitergehende Anzeigen und Konfigurationen ähnlich XP mit
 - ipconfig /all
 - route -6 print
 - ping -6
 - tracert -6
 - netsh interface ipv6 show address
 - netsh interface ipv6 show interface
 - netsh interface ipv6 show route
 - netsh interface teredo show state
 - nslookup -q=any <dns-name>
- Hilfestellung im Microsoft TechNet:
 - <http://technet.microsoft.com/en-us/library/bb742622.aspx>
 - <http://technet.microsoft.com/en-us/network/bb530961.aspx>



Windows 7 – ipconfig

```
C:\Users\wessend>ipconfig
Windows-IP-Konfiguration

Ethernet-Adapter LAN-Verbindung:
  Verbindungsspezifisches DNS-Suffix: 2001:638:500:2500:40fe:906f:7db5:56d9
  Temporäre IPv6-Adresse: 2001:638:500:2500:7004:cafe:93be:b77e
  Verbindungslokale IPv6-Adresse: fe80::40fe:906f:7db5:56d9%9
  IPv4-Adresse: 128.176.190.51
  Subnetzmaske: 255.255.255.192
  Standardgateway: fe80::208:e2ff:fe0e:2008%9
                           128.176.190.1

Tunneladapter Teredo Tunneling Pseudo-Interface:
  Medienstatus: Medium getrennt
  Verbindungsspezifisches DNS-Suffix: 2001:638:500:2500:40fe:906f:7db5:56d9

Tunneladapter isatap.<2EDBE4FB-3F0A-4FCA-BA33-2E017083425E>:
  Medienstatus: Medium getrennt
  Verbindungsspezifisches DNS-Suffix: 2001:638:500:2500:40fe:906f:7db5:56d9

Tunneladapter Reusable ISATAP Interface <26B0DE20-CAAE-4FFC-B21C-500885418537>:
  Medienstatus: Medium getrennt
  Verbindungsspezifisches DNS-Suffix: 2001:638:500:2500:40fe:906f:7db5:56d9

C:\Users\wessend>
```

Windows 7 – route print -6

```
C:\Users\wessend>route print -6
=====
Schnittstellenliste
  9...00 0c 29 54 17 6f ....VMware Accelerated AMD PCNet Adapter
  1.....Software Loopback Interface 1
  16...00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
  10...00 00 00 00 00 00 e0 Microsoft-ISATAP-Adapter
  11...00 00 00 00 00 00 e0 Microsoft-ISATAP-Adapter #2
=====

IPv6-Routentabelle
=====
Aktive Routen:
  If Metrik Netzwerkziel           Gateway
  9   266 ::/0                      fe00::208:e0ff:fe0e:2008
  1   306 ::1/128                   Auf Verbindung
  9   18 2001:638:500:2500::/64    Auf Verbindung
  9   266 2001:638:500:2500:40fe:906f:7db5:56d9/128
  9   266 2001:638:500:2500:7004:cafe:93be:b77e/128
  9   266 fe80::/64                Auf Verbindung
  9   266 fe80::40fe:906f:7db5:56d9/128
  1   306 ff00::/8                 Auf Verbindung
  9   266 ff00::/8                 Auf Verbindung
=====

Ständige Routen:
  Keine
```

Windows 7 – ping

```
C:\Users\wessend>ping -4 www.heise.de
Ping wird ausgeführt für www.heise.de [193.99.144.85] mit 32 Bytes Daten:
Antwort von 193.99.144.85: Bytes=32 Zeit=20ms TTL=241
Antwort von 193.99.144.85: Bytes=32 Zeit=19ms TTL=241
Antwort von 193.99.144.85: Bytes=32 Zeit=19ms TTL=241
Antwort von 193.99.144.85: Bytes=32 Zeit=20ms TTL=241

Ping-Statistik für 193.99.144.85:
  Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
  (0% Verlust).
Ca. Zeitangaben in Millisek.:
  Minimum = 19ms, Maximum = 20ms, Mittelwert = 19ms

C:\Users\wessend>ping -6 www.heise.de
Ping wird ausgeführt für www.heise.de [2a02:2e0:3fe:100::7] mit 32 Bytes Daten:
Antwort von 2a02:2e0:3fe:100::7: Zeit=20ms
Antwort von 2a02:2e0:3fe:100::7: Zeit=19ms
Antwort von 2a02:2e0:3fe:100::7: Zeit=19ms
Antwort von 2a02:2e0:3fe:100::7: Zeit=18ms

Ping-Statistik für 2a02:2e0:3fe:100::7:
  Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
  (0% Verlust).
Ca. Zeitangaben in Millisek.:
  Minimum = 18ms, Maximum = 20ms, Mittelwert = 19ms

C:\Users\wessend>ping www.heise.de
Ping wird ausgeführt für www.heise.de [2a02:2e0:3fe:100::7] mit 32 Bytes Daten:
Antwort von 2a02:2e0:3fe:100::7: Zeit=19ms
Antwort von 2a02:2e0:3fe:100::7: Zeit=19ms
Antwort von 2a02:2e0:3fe:100::7: Zeit=19ms
Antwort von 2a02:2e0:3fe:100::7: Zeit=19ms

Ping-Statistik für 2a02:2e0:3fe:100::7:
  Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
  (0% Verlust).
Ca. Zeitangaben in Millisek.:
  Minimum = 19ms, Maximum = 19ms, Mittelwert = 19ms
```

Windows 7 – netsh interface ipv6

```
C:\Users\wessend>netsh int ipv6
Folgende Befehle sind verfügbar:
Befehle in diesen Kontext:
6to4      - Wechselt zum 'netsh interface ipv6 6to4'-Kontext.
?         - Zeigt eine Liste der Befehle an.
add       - Fügt einen Konfigurationseintrag zu einer Tabelle hinzu.
delete    - Löscht einen Konfigurationseintrag aus einer Tabelle.
dump      - Zeigt ein Konfigurationsskript an.
help      - Zeigt eine Liste der Befehle.
isatap    - Wechselt zum 'netsh interface ipv6 isatap'-Kontext.
reset    - Setzt die IP-Konfigurationen zurück.
set      - Legt Konfigurationsinformationen fest.
show     - Zeigt Informationen an.

Folgende Unterkontakte sind verfügbar:
6to4 isatap

Geben Sie den Befehl, gefolgt von einem Leerzeichen und ? ein, um Hilfe bezüglich des entsprechenden Befehls zu erhalten.

C:\Users\wessend>netsh int ipv6 show
Folgende Befehle sind verfügbar:
Befehle in diesen Kontext:
show addresses - Zeigt die aktuellen IP-Adressen an.
show compartments - Zeigt die Depotparameter an.
show destinationcache - Zeigt Zielcacheeinträge an.
show dnsservers - Zeigt die aktiven DNS-Adressen an.
show dynamicportrange - Zeigt Konfigurationsparameter für dynamischen Portbereich an.
show global    - Zeigt globale Konfigurationsparameter an.
show interfaces - Zeigt Schnittstellenparameter an.
show ipstats   - Zeigt IP-Statistiken an.
show linklayer - Zeigt die aktiven Multicastgruppen an.
show neighbors - Zeigt Nachbarcacheeinträge an.
show offload   - Zeigt die Abladeinformationen an.
show potentialrouters - Zeigt potentielle Routen an.
show prefixpolicies - Zeigt Prefix-Richtlinieneinträge an.
show proxy     - Zeigt den aktiven Proxyparameter an.
show route    - Zeigt Routentabelleinträge an.
show siteprefixes - Zeigt Standortprefix-Tabelleinträge an.
show tapstats - Zeigt TCP-Statistiken an.
show teredo   - Zeigt den Teredo-Status an.
show udpstats - Zeigt UDP-Statistiken an.
```

Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

79



Host-Beispiel: Linux

- seit Kernel 2.1.8 ist IPv6-Code integriert
 - heute unterstützen alle gängigen Linux-Distributionen IPv6
- Kommando-Beispiele:
 - `/sbin/ifconfig -a` (zeigt alle Interfaces an)
 - `/sbin/route -A inet6 -n` (zeigt Routingtabelle an)
 - `netstat -A inet6 -r -n` “
 - `/sbin/ip -f inet6 neigh` (zeigt Neighbor-Cache an)
 - `nslookup -q=any <dns-name>` (fragt Nameserver ab)
- Konnektivitäts-Tests:
 - `/bin/ping6`
 - `/bin/traceroute6`
- die meisten Applikationen sind IPv6-fähig ...

Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

80



Linux – Kommando-Beispiele

```
[root@gucky ~]# ifconfig eth0.2500
eth0.2500 Link encap:Ethernet HWaddr 00:1E:C9:55:05:89
      inet addr:128.176.190.56 Bcast:128.176.190.63 Mask:255.255.255.192
        inet6 addr: fe80::21e:c9ff:fe55:589/64 Scope:Global
           inet6 addr: fe80::21e:c9ff:fe55:589/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:1399 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1089 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:349302 (341.1 KiB) TX bytes:422812 (412.9 KiB)

[root@gucky ~]# route -A inet6 | grep eth0.2500
2001:638:500:2500::/64          *
                           *          UA 256 0 0 eth0.2500
fe80::/64                      *          U 256 0 0 eth0.2500
*/*0                            fe80::208:e2ff:fe0e:2008  UGDA 1024 1 0 eth0.2500
ff02::d/128                      ff02::d          UC 0 4 0 eth0.2500
ff00::/8                         *          U 256 0 0 eth0.2500

[root@gucky ~]# traceroute -n ipv6.google.com
traceroute to ipv6.google.com (2a00:1450:8007::6a), 30 hops max, 40 byte packets
 1  2001:638:500:200:208:e2ff:fe0e:2008  0.966 ms  1.103 ms  1.253 ms
  2  2001:638:c:ao02c::1  0.700 ms  0.729 ms  0.829 ms
  3  2001:638:c::034::1  2.874 ms  2.976 ms  3.077 ms
  4  2001:638:c::030::1  6.016 ms  6.027 ms  6.008 ms
  5  2001:638:c::004::1  9.869 ms  9.891 ms  9.873 ms
  6  2001:778:19:1:3b41:1  10.316 ms  10.281 ms  10.266 ms
  7  2001:4860::1:0:60d  10.870 ms  10.914 ms  2001:4860::1:0:5bd  10.739 ms
  8  2001:4860::1:0:fbc  16.699 ms  2001:4860::1:0:fd  20.934 ms  2001:4860::1:0:fbc  16.605 ms
  9  2001:4860::1:0:10  24.520 ms  2001:4860::1:0:11  21.660 ms  2001:4860::1:0:10  21.533 ms
 10  2001:4860::2:0:48d  21.779 ms  21.619 ms  21.532 ms
 11  2001:4860::1:c9  29.845 ms  2001:4860::1:c7  30.683 ms  29.814 ms
 12  2a00:1450:8007::6a  22.523 ms  21.793 ms  22.061 ms
```

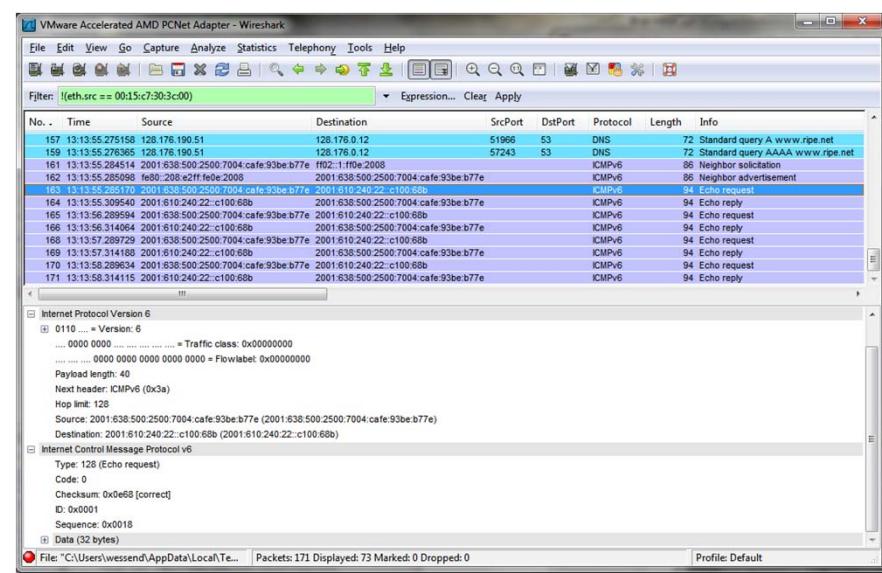
Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

81

ZENTRUM FÜR
INFORMATION
VERARBEITUNG

IPv6 im Protokollanalyse

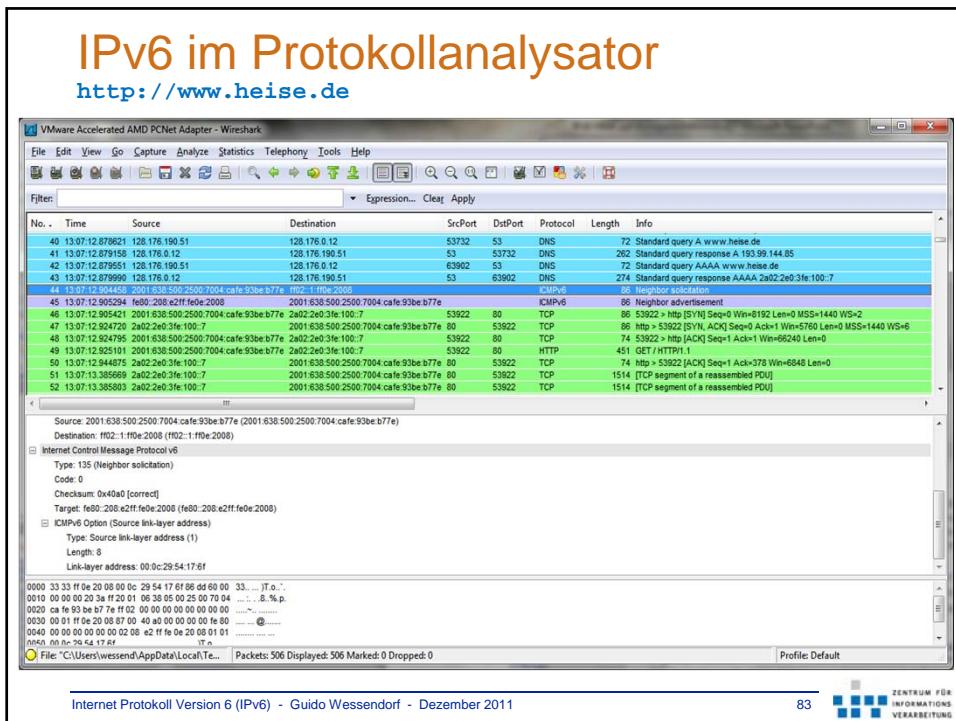
ping www.ripe.net



Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

82

ZENTRUM FÜR
INFORMATION
VERARBEITUNG



Literatur / Links

- einige Basis-Spezifikationen
 - [RFC 2460: Internet Protocol, Version 6 \(IPv6\) Specification](#)
 - [RFC 4861: Neighbor Discovery for IP Version 6 \(IPv6\)](#)
 - [RFC 4862: IPv6 Stateless Address Autoconfiguration](#)
 - [RFC 4311: IPv6 Host-to-Router Load Sharing](#)
 - [RFC 4443: Internet Control Message Protocol \(ICMPv6\) for the IPv6 Specification](#)
 - [RFC 2464: Transmission of IPv6 Packets over Ethernet Networks](#)
 - [RFC 4291: Internet Protocol Version 6 \(IPv6\) Addressing Architecture](#)
 - [RFC 3587: An IPv6 Aggregatable Global Unicast Address Format](#)
- einige Web-Links
 - <http://en.wikipedia.org/wiki/IPv6>
 - <http://www.ipv6forum.com/>
 - <http://www.6diss.org/>
 - <http://www.ipv6.com/>
 - <http://www.ipv6tf.org/>
 - <http://www.6net.org/>
- einige über IPv6 erreichbare Web-Seiten
 - <http://ipv6.google.com/>
 - <http://www.ripe.net/>
 - [http://www.heise.de/ \(IPv6/IPv4\)](http://www.heise.de/), [http://www.six.heise.de/ \(IPv6-only\)](http://www.six.heise.de/)

Internet Protokoll Version 6 (IPv6) - Guido Wessendorf - Dezember 2011

Vielen Dank für die Aufmerksamkeit!
Fragen, Anregungen, Kritik?

Guido Wessendorf
Westfälische Wilhelms-Universität Münster
Zentrum für Informationsverarbeitung
Röntgenstrasse 7-13
48149 Münster
wessend@uni-muenster.de

