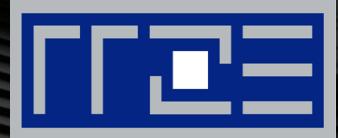


# REGIONALES RECHENZENTRUM ERLANGEN [RRZE]



## IP-FAU-6 Teil 2 – IPv6 am Endgerät

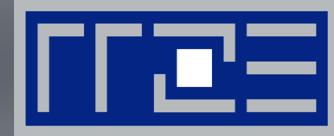
Netzwerkausbildung – Praxis der Datenkommunikation  
09.12.2015, Jochen Reinwand, Holger Marquardt, RRZE

# Gliederung

- Adressen
  - Adresstypen, Adresszuweisung
  - Betriebssysteme, Anwendungen
- Linux / SUSE
  - YaST, Systemtools
- Windows
  - Konfiguration und Diagnose
- Datenschutz und Privatsphäre
- Gefahr durch den IPv6-Tunnel Teredo?
  - Teredo-Adressen
  - Verbindungsaufbau
  - Nachteile und Gefahren
- Stabilität und Sicherheit



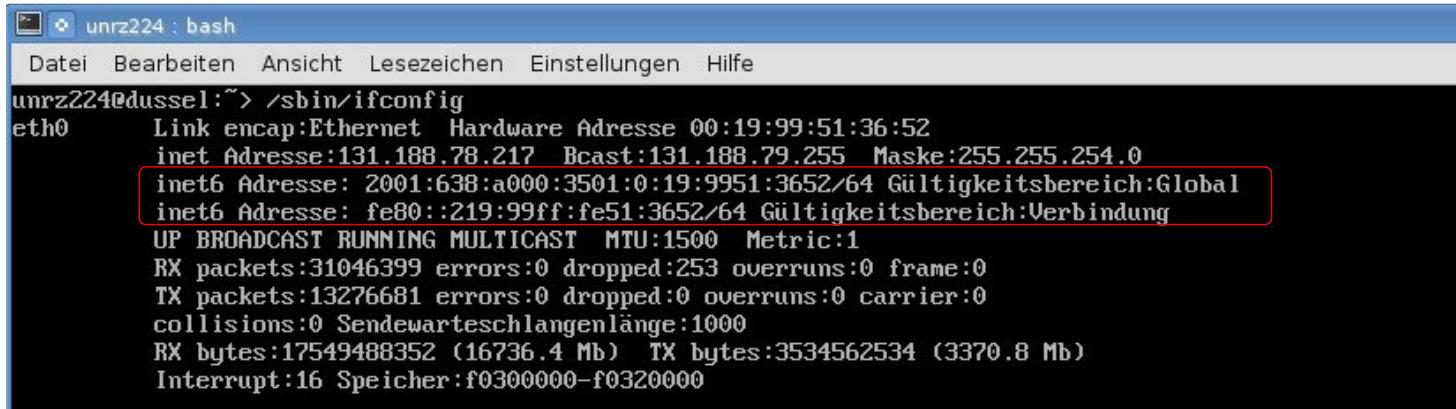
# ADRESSEN



Adresstypen, Adresszuweisung  
Betriebssysteme, Anwendungen

# IPv6-Adresstypen

Blick auf die Schnittstellenkonfiguration eines normalen PC mit IPv6-Zugang („ifconfig“ (Linux/Unix) bzw. „ipconfig“ (Windows))



```
unrz224@dussel:~> /sbin/ifconfig
eth0      Link encap:Ethernet  Hardware Adresse 00:19:99:51:36:52
          inet  Adresse:131.188.78.217  Bcast:131.188.79.255  Maske:255.255.254.0
                   inet6 Adresse: 2001:638:a000:3501:0:19:9951:3652/64  Gültigkeitsbereich:Global
                   inet6 Adresse: fe80::219:99ff:fe51:3652/64  Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31046399  errors:0  dropped:253  overruns:0  frame:0
          TX packets:13276681  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  Sendewarteschlangenlänge:1000
          RX bytes:17549488352 (16736.4 Mb)  TX bytes:3534562534 (3370.8 Mb)
          Interrupt:16  Speicher:f0300000-f0320000
```

IPv6-  
konfigurierter  
Linux-Rechner

→ Überraschend: **Zwei IPv6-Adressen!**

- › 2001:638:a000:3501:0:19:9951:3652/64 ist die „normale“ Adresse
- › fe80::219:99ff:fe51:3652/64 ist die sog. „Link-Local“-Adresse

# Link-Local-Adressen (fe80::)

- Auch bekannt als „verbindungslokale Adressen“
  - fangen immer mit fe80: an
  - kann der normale User getrost ignorieren ;)
  - vom Betriebssystem automatisch für jedes Interface generiert (MAC)
  - „einfach da“, egal ob das restliche Netz IPv6 hat oder nicht
  - verlassen niemals das Subnetz (keine „normale“ Kommunikation)
  - sollte man nie ohne Not deaktivieren
- Zweck
  - Grundlage für das Neighbor Discovery Protocol (NDP)
  - Das grundlegende Management-Subsystem des IPv6-Protokolls
  - Entspricht dem „ARP“-Protokoll unter IPv4, kann aber mehr
  - Bsp: Adress-Resolution, Nachbarerkennung, Routersuche, Adresskonflikte,...

# Neighbor Discovery Protocol (NDP)

- Basiert auf ICMPv6
- Diagnostische Möglichkeiten am Beispiel Linux
  - funktioniert dank Link-Local-Adressen auch ohne konfiguriertes IPv6
  - ndisc6/rdisc6: <http://www.remlab.net/ndisc6/>
- „Alle bekannten IPv6-Knoten im Netzwerksegment anzeigen“
  - ip -6 neigh
  - Apple: ndp
  - Windows: netsh interface ipv6 show neighbors
- „MAC-Adresse zu v6-Adresse rausfinden“ (entsp. IPv4 ARP-Lookup)
  - ndisc6 2001:638:a000:1021:21::1 eth0
- „Router im Netz suchen und Netzpräfix geben lassen“
  - rdisc6 eth0

# Neighbor Discovery Protocol (NDP)

- „Einen bestimmten Knoten im Netzwerksegment anpingen“  
> `ping6 -I eth0 fe80::2d0:1ff:fef7:ac00`  
Oder:  
> `ping6 fe80::2d0:1ff:fef7:ac00%eth0`
- „Alle Knoten im gleichen Netzwerksegment anpingen“  
(→ Multicast)  
> `ping6 -I eth0 ff02::1`
- „Alle Router im gleichen Netzwerksegment anpingen“  
(→ Multicast)  
> `ping6 -I eth0 ff02::2`

# Adresszuweisung – Manuelle Konfiguration

- Durch Automatismen und Vorgaben deutlich vereinfacht
- Nötige Angaben: IP-Adresse, (DNS-Server)
- In der Regel entfällt:
  - Netzmaske → *Router-Advertisment (RA)* – Wieder NDP
  - Broadcast → Multicast
  - Gateway → *Router-Advertisment (RA)* – Wieder NDP

# Adresszuweisung – SLAAC

- *Stateless Address AutoConfiguration*, RFC 4862
- Auch auf Basis von NDP
- Link-Local und globale Adresse werden von Host selbst vergeben und per Multicast-Verfahren verifiziert
  - MAC-Adresse: 00:19:99:51:36:52
  - EUI64-Format: 02:19:99:ff:fe:51:36:52, bzw 0219:99ff:fe51:3652
  - Link-Local-Adresse: fe80::219:99ff:fe51:3652/64
  - Globale Adresse: <Präfix>:219:99ff:fe51:3652/64
- Probleme
  - *Stateless* → Keine statische Vorgabe von Adressen!
  - Nur begrenzte Menge an Informationsparametern
  - Ungeeignet für „Managed Networks“ (wie FAU)

# Adresszuweisung – DHCPv6

- *Dynamic Host Configuration Protocol*
- „Stateful Address Autoconfiguration“
- Praktisch äquivalent zu IPv4
- Aber: Keine „Konkurrenz-Optionen“ zu NDP
- Zuerst Benutzung der MAC-Adresse nicht erlaubt!
  - DUID (DHCP Unique Identifier)
  - Lösung: Moderne OS bilden DUID auf Basis von MAC und DHCP-Server verwenden letztere statt willkürlicher DUID
  - **Vorsicht:** Ein Client hat oft nur eine DUID, die auf Basis eines Interfaces gebildet wird! Diese wird dann für alle Interfaces verwendet. Leider gilt dies beispielsweise für Windows.

# Adresszuweisung – DHCPv6

- Referenzimplementierung ISC dhcpcd ist Beispiel für Dienst mit problematischer Implementierung von Dual-Stack
- An der FAU: Kein SLAAC → Statisch oder DHCPv6
  - Kontakt wie bei IPv4: [dhcp@fau.de](mailto:dhcp@fau.de)

# Übersicht Betriebssysteme

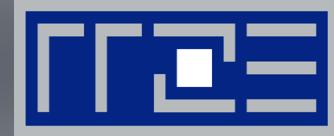
- Nahezu alle aktuellen Betriebssysteme unterstützen IPv6
  - Mac OS X: ab 10.2 enthalten, ab 10.3 per GUI konfigurierbar
  - Verschiedenste UNIX-Varianten (AIX, BSD, HP-UX, Solaris...)
  - Router (Cisco, Juniper)
  - Windows:
    - ab XP oder Server 2003 mit aktuellem Service Pack
    - Volle Unterstützung ("Dual-IP-Layer-Architektur"): Windows Vista, Windows Server 2008 und höher
  - Linux: Grundsystem schon lange IPv6-fähig  
Aber: Konfigurationstools und GUI (Graphical User Interface) des jeweiligen Anbieters (Distribution) wichtig
- Im Folgenden näher betrachtet: Linux und Windows

# Anwendungen

- Die meisten Anwendungen sind mittlerweile IPv6-fähig
- Manchmal ist es durchaus etwas hackelig, z.B. bei Eingabe von URLs wegen nicht eindeutiger „:“
- Dank DNS für den normalen Nutzer auf Anwendungsebene in der Regel transparent
- Aber: Viele Webseiten haben beispielsweise noch keine IPv6-Adresse, da durch den Vorrang von IPv6 zu viele Probleme erwartet werden.  
Zwei weltweite Aktionen, denen eine Aktion des Heise-Verlags als Vorbild diente:
  - World IPv6 Day (2011, Test, <http://www.worldipv6day.org/>)
  - World IPv6 Launch Day (2012, dauerhafte Umschaltung, <http://www.worldipv6launch.org/>)



# LINUX / SUSE

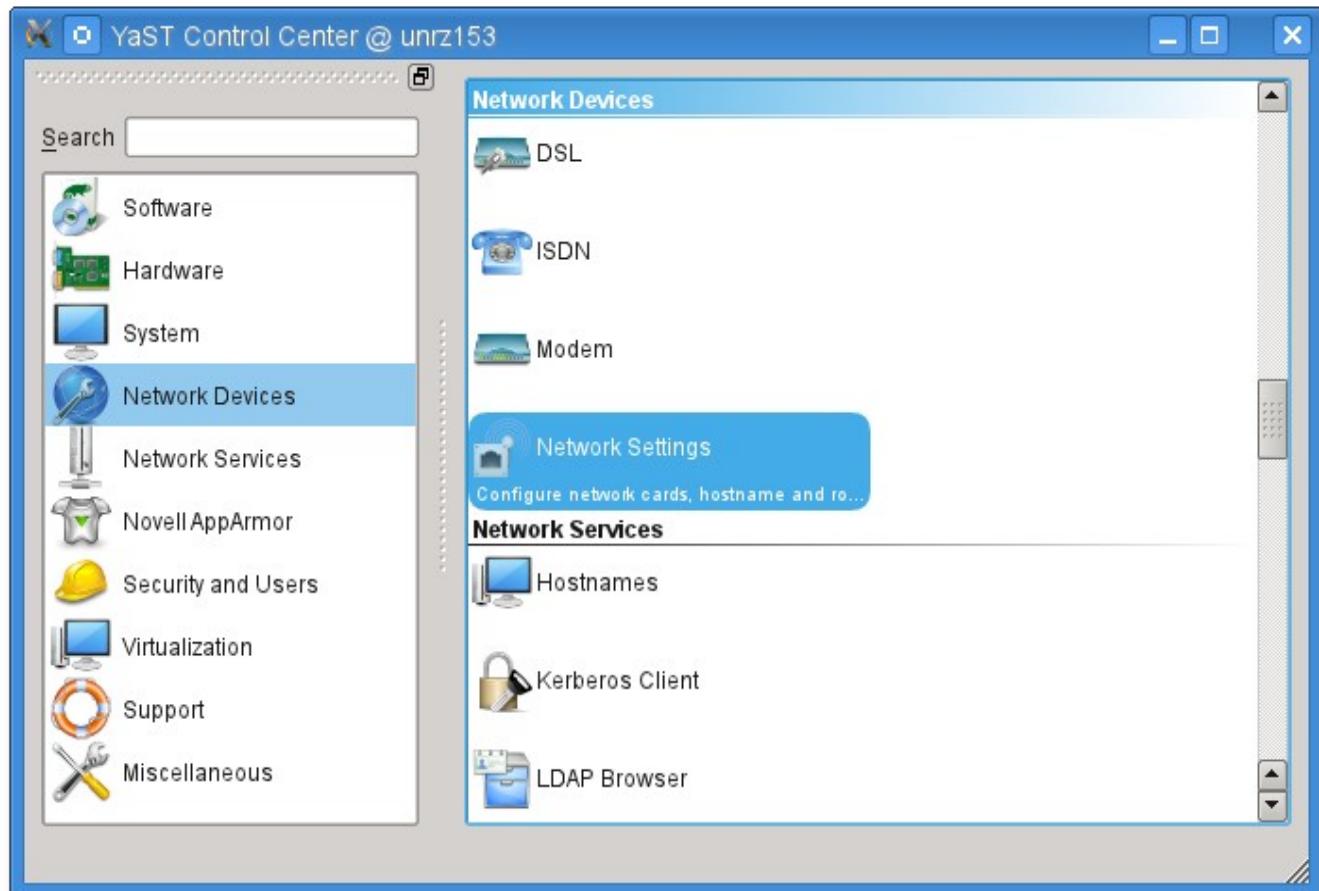


Yast, Systemtools

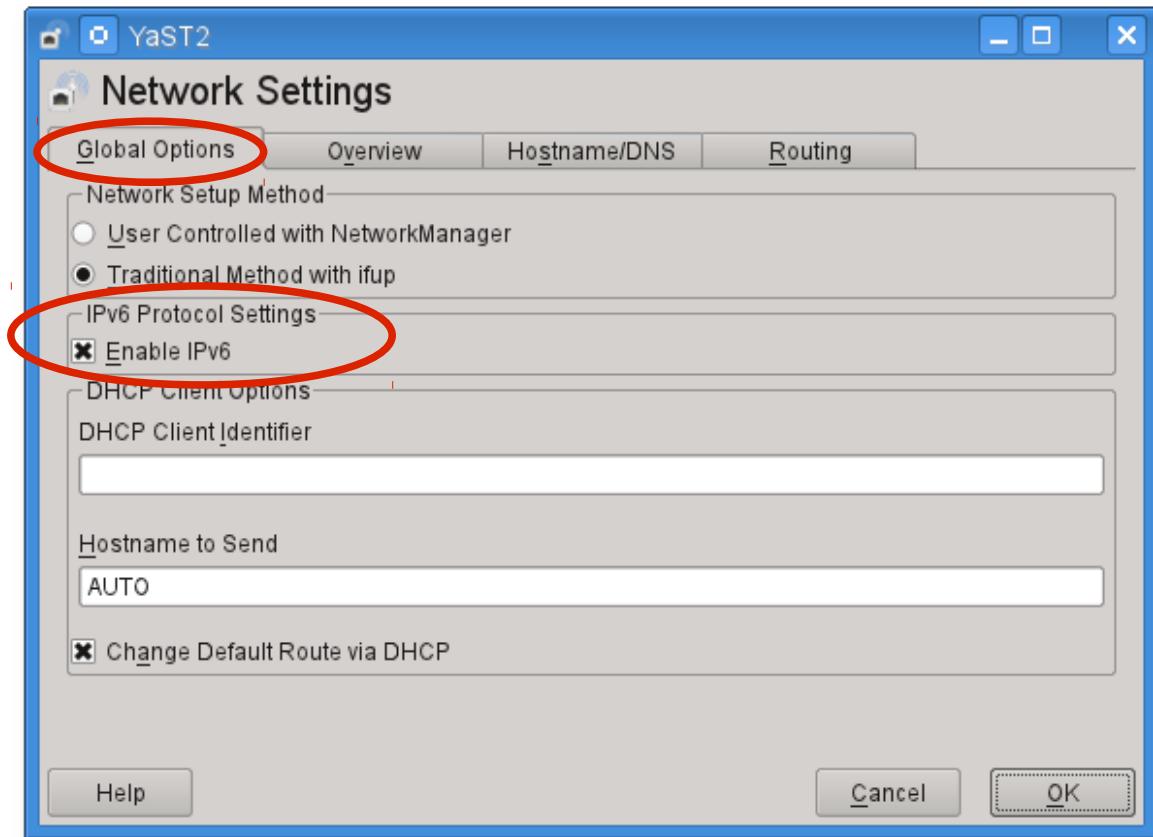
# Linux / SUSE

- Grundlegende Tools unter den meisten Linux-Distributionen identisch und entsprechen denen für IPv4 (ifconfig, route, ip usw.)
- Konfiguration unter SUSE wie gewohnt via YaST2  
IPv4 und IPv6 dabei gleichwertig integriert
- Im Hintergrund als „Zwischenschicht“:  
ifup, NetworkManager, wicd, ...
- Wenn es doch mal Probleme gibt:  
Wie schaltet man IPv6 vollständig ab?
  - Nur durch Entfernen des Kernel-Moduls!
  - Die meisten Distributionen bieten eine Deaktivierung zwar per Mausklick, aber Reboot ist zumeist dennoch notwendig/empfohlen!

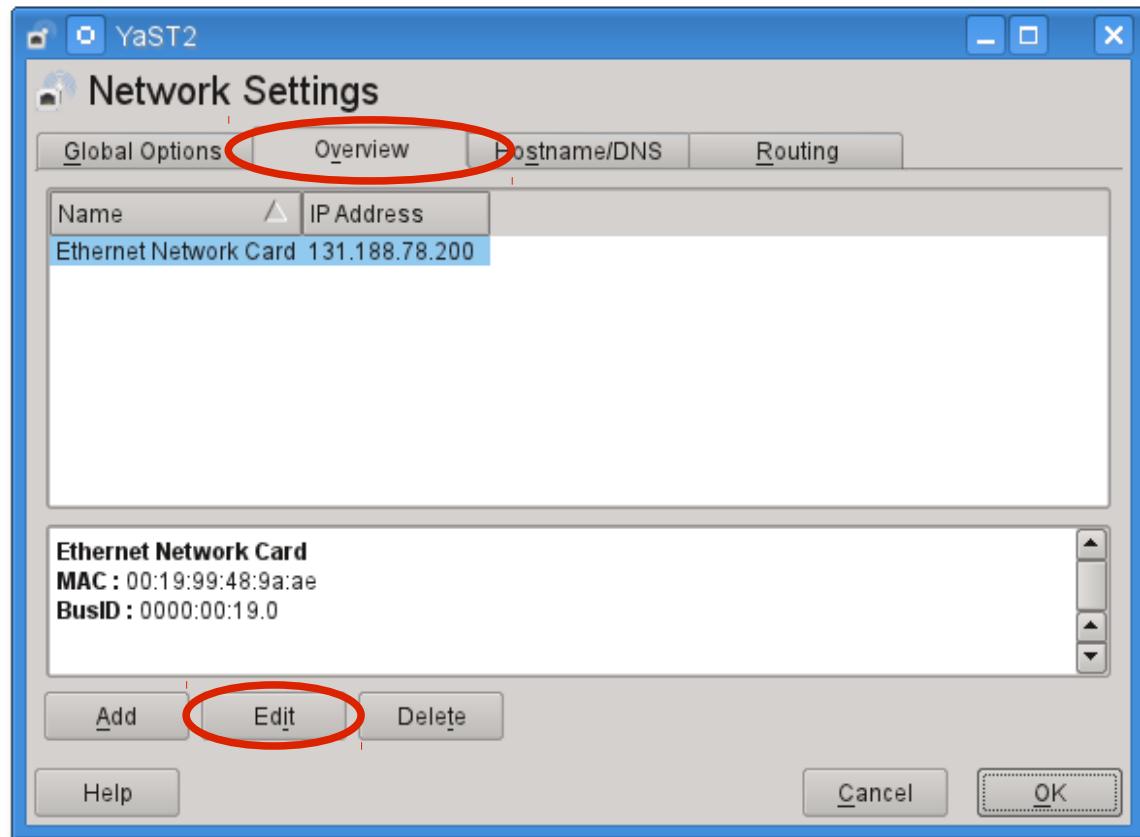
# Linux / SUSE YaST



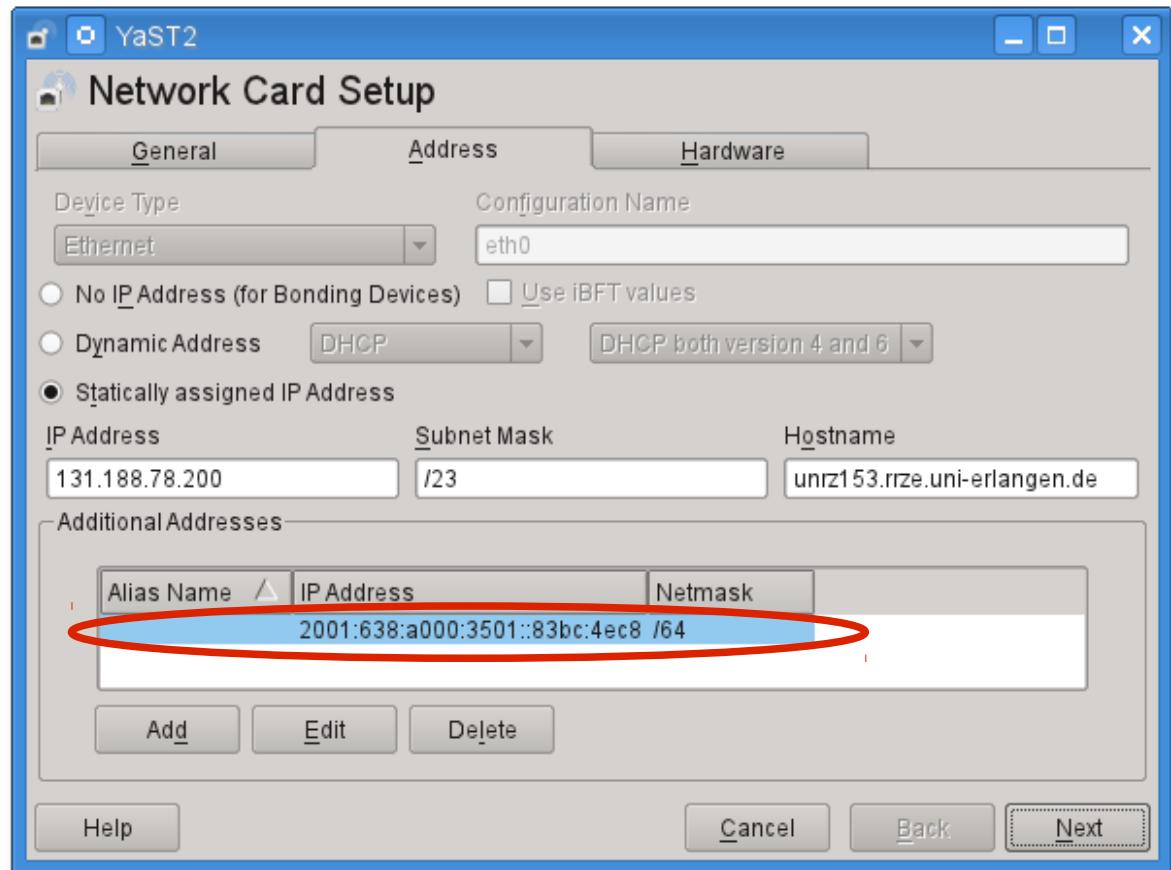
# Linux / SUSE YaST



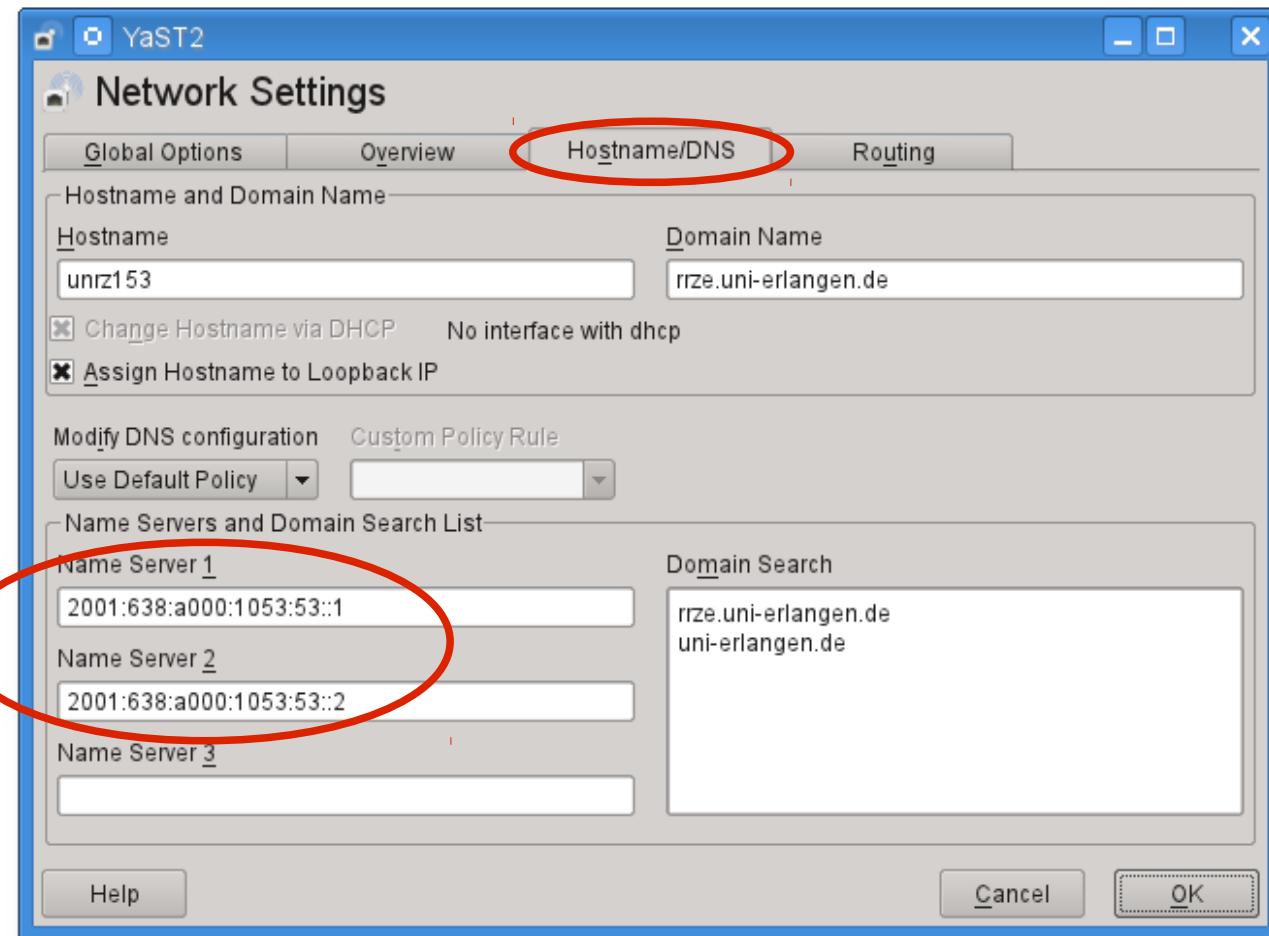
# Linux / SUSE YaST



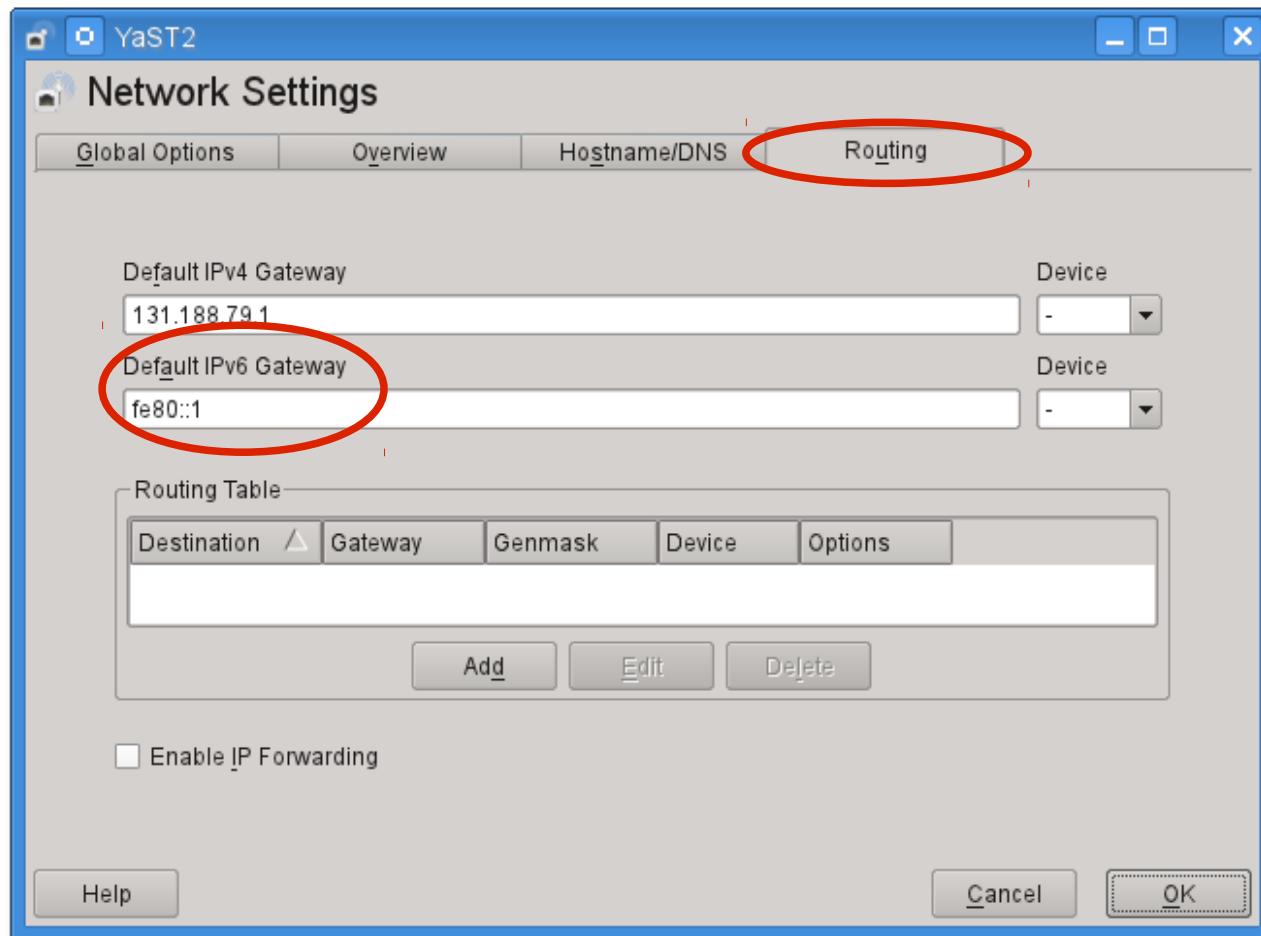
# Linux / SUSE YaST



# Linux / SUSE YaST



# Linux / SUSE YaST



# Linux / SUSE – Systemtools

- Allgemeine Linux-Tools sind weiterhin im Hintergrund
- Oft bedienen sie beide Protokolle gleichzeitig:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:19:99:48:9A:AE
          inet  addr:131.188.78.200  Bcast:131.188.79.255  Mask:255.255.254.0
                  inet6 addr: 2001:638:a000:3501::83bc:4ec8/64  Scope:Global
                  inet6 addr: fe80::219:99ff:fe48:9aae/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22916995 errors:0 dropped:189 overruns:0 frame:0
          TX packets:8749208 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14215281746 (13556.7 Mb)  TX bytes:2638188558 (2515.9 Mb)
          Interrupt:23  Memory:d0020000-d0040000
```

- Merke: Mehrere IP-Adressen pro Interface sind unter IPv4 die Ausnahme und unter IPv6 die Regel!

# Linux / SUSE – Systemtools

- Manchmal muss man IPv6 explizit anfordern:

```
# route --inet6 -n
Kernel IPv6 routing table
Destination          Next Hop    Flags Metric Ref Use Iface
2001:638:a000:3501::/64      ::      U    256   9643  0  eth0
fe80::/64            ::      U    256   0      0  eth0
::/0                 fe80:::1  UGDA 1024  122194 6  eth0
::1/128             ::      U    0     23095  7  lo
2001:638:a000:3501::83bc:4ec8/128  ::      U    0     1992198 1  lo
fe80::219:99ff:fe48:9aae/128  ::      U    0     35028  1  lo
ff02::1/128          ff02::1  UC    0     1      0  eth0
ff02::fb/128          ff02::fb UC    0     795   0  eth0
ff00::/8             ::      U    256   0      0  eth0
```

# Linux / SUSE – Systemtools

- Manchmal geschieht Auswahl mit angehänger 6  
Aber: Unterschiedlich implementiert!
- ping und ping6 sind getrennte Programme  
→ Auswahl "von Hand"
- traceroute6 ist Symlink auf traceroute Programm  
→ Ein Programm, das beide Protokolle spricht  
→ IPv6 traceroute sobald IPv6-Adresse vorhanden  
Zusätzlich Parameter -4 und -6 um jeweilige Version zu  
erzwingen

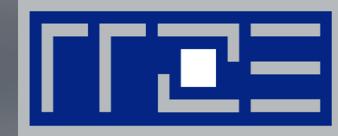
# Linux / SUSE – Systemtools

```
# traceroute dns1.rrze.uni-erlangen.de
traceroute to dns1.rrze.uni-erlangen.de (2001:638:a000:1053:53::1), 30 hops max, 40 byte packets using UDP
1  sitak.gate.uni-erlangen.de (2001:638:a000:3501::2)  17.168 ms   16.107 ms   14.930 ms
2  constellation.gate.uni-erlangen.de (2001:638:a000::3336:33)  1.654 ms   1.583 ms   1.279 ms
3  prime.gate.uni-erlangen.de (2001:638:a000::333b:3b)  0.822 ms   0.776 ms   0.762 ms
4  dns1.rrze.uni-erlangen.de (2001:638:a000:1053:53::1)  0.336 ms   0.369 ms   0.299 ms

# traceroute -4 dns1.rrze.uni-erlangen.de
traceroute to dns1.rrze.uni-erlangen.de (131.188.0.10), 30 hops max, 40 byte packets using UDP
1  131.188.79.253 (131.188.79.253)  4.622 ms   3.599 ms   2.468 ms
2  constellation.gate.uni-erlangen.de (131.188.20.201)  1.133 ms   0.742 ms   0.646 ms
3  prime.gate.uni-erlangen.de (131.188.10.34)  0.462 ms   0.406 ms   0.397 ms
4  dns1.rrze.uni-erlangen.de (131.188.0.10)  0.208 ms   0.213 ms   0.196 ms
```



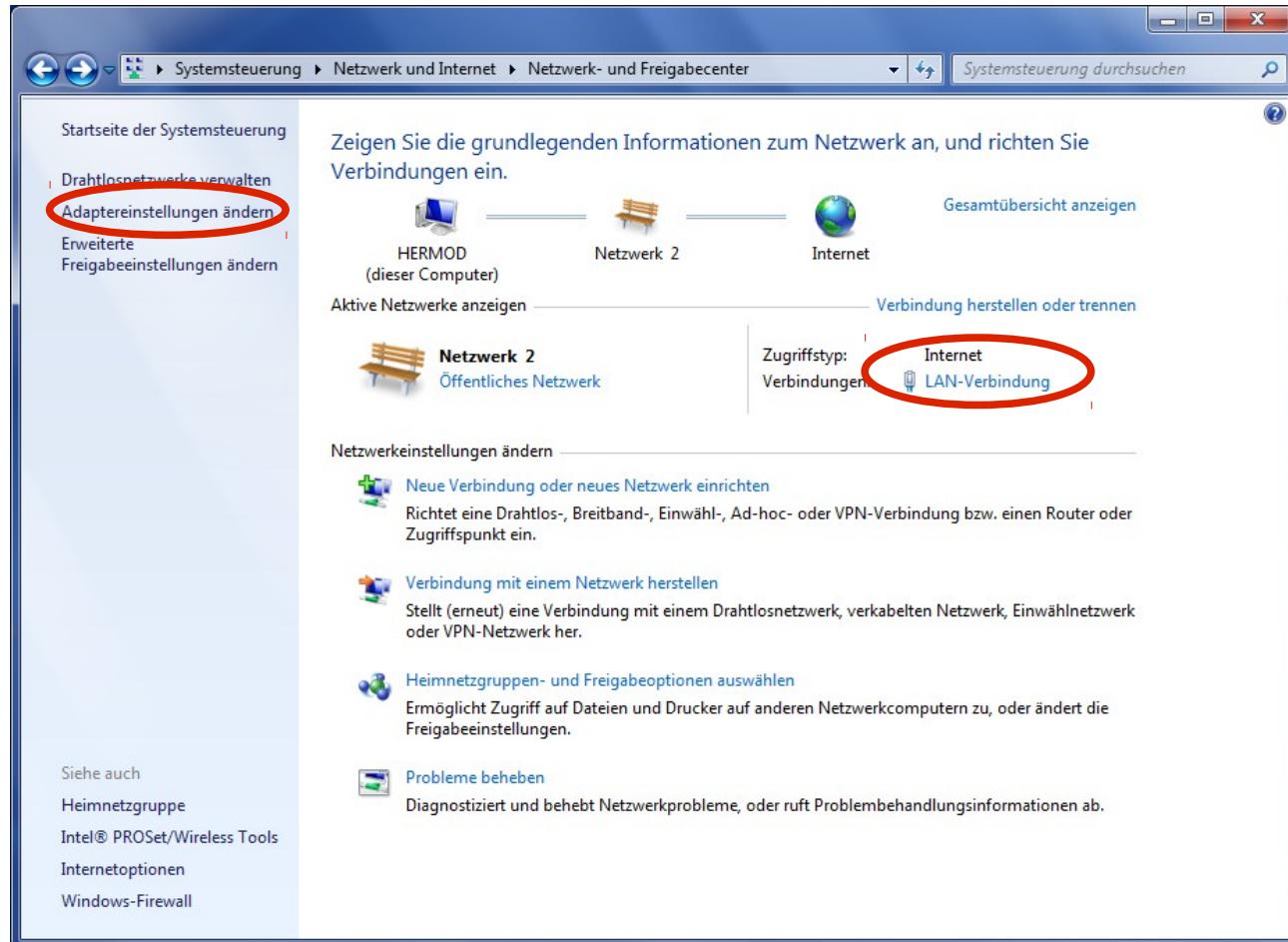
# WINDOWS



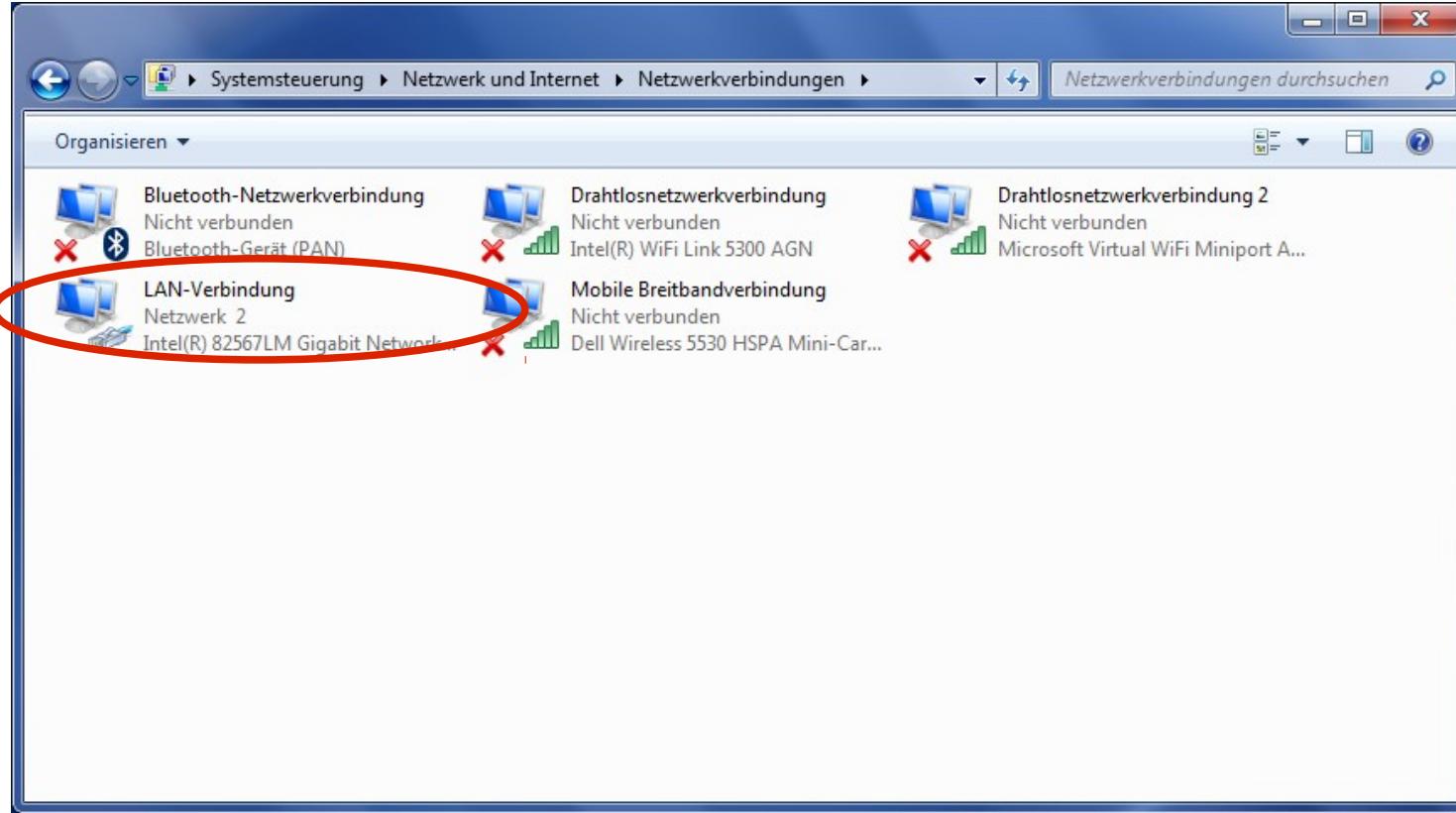
Konfiguration und Diagnose unter Windows  
Vista und später

# Windows Konfiguration

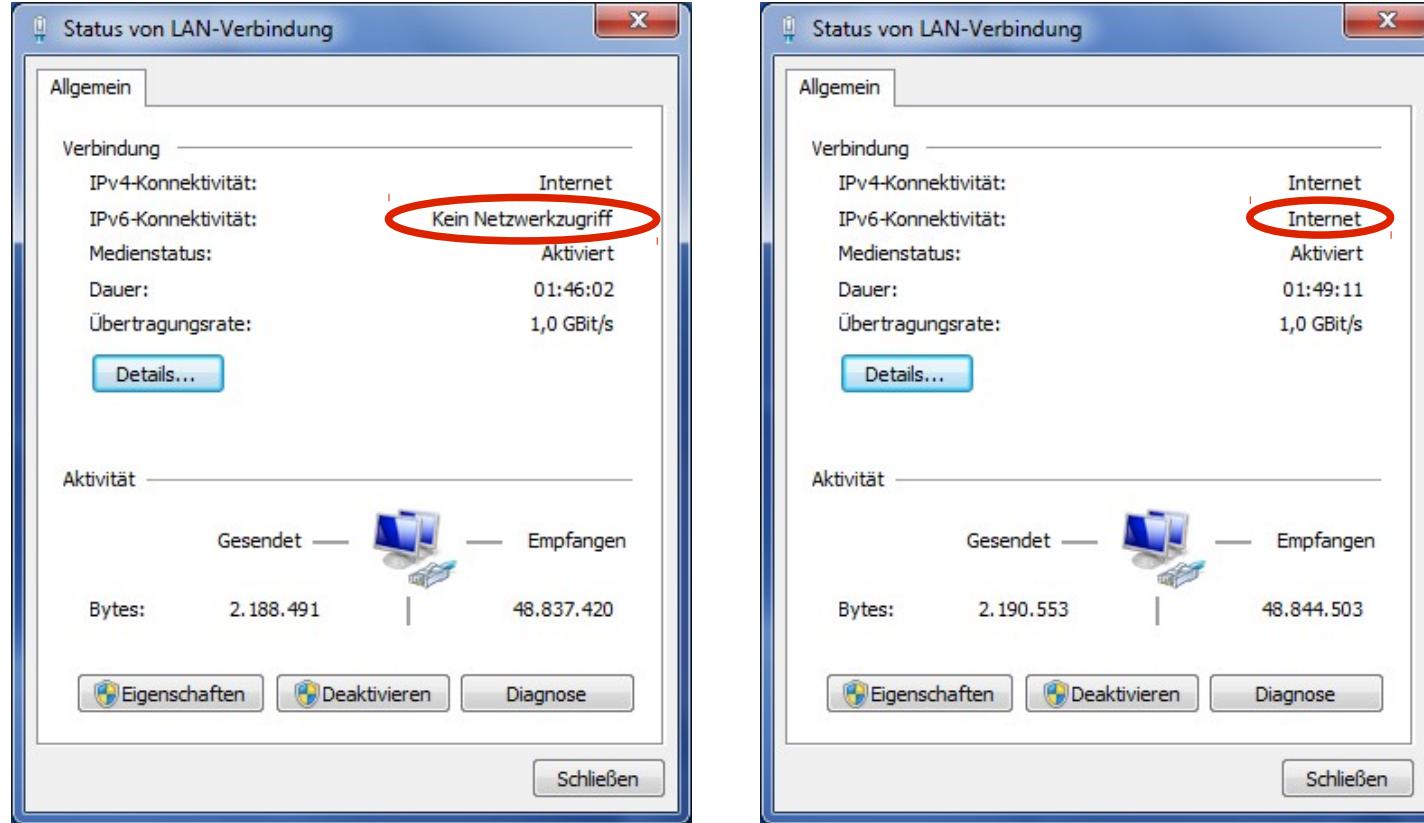
- Ab Windows 7  
IPv6-Support  
„vollständig“



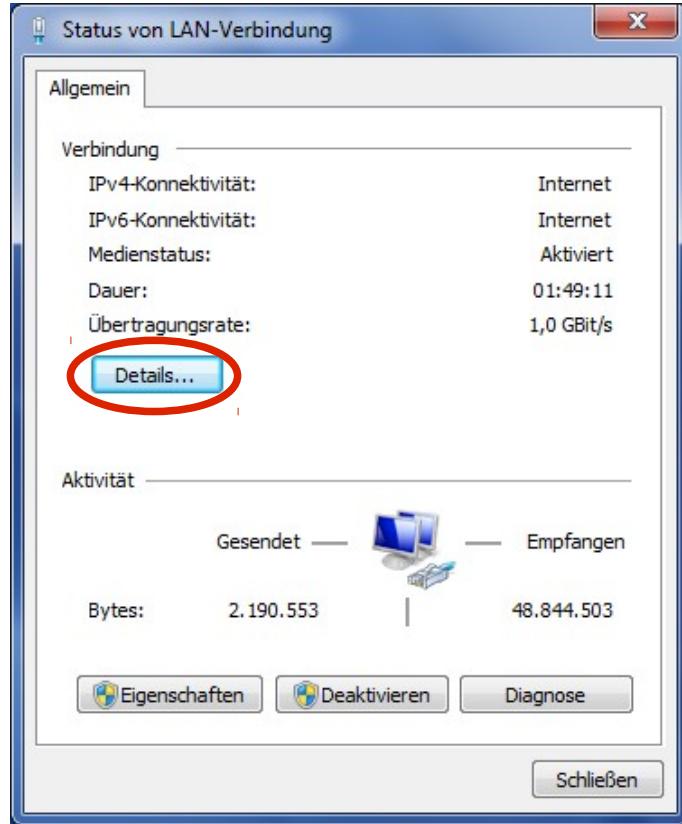
# Windows – Konfiguration



# Windows – Konfiguration

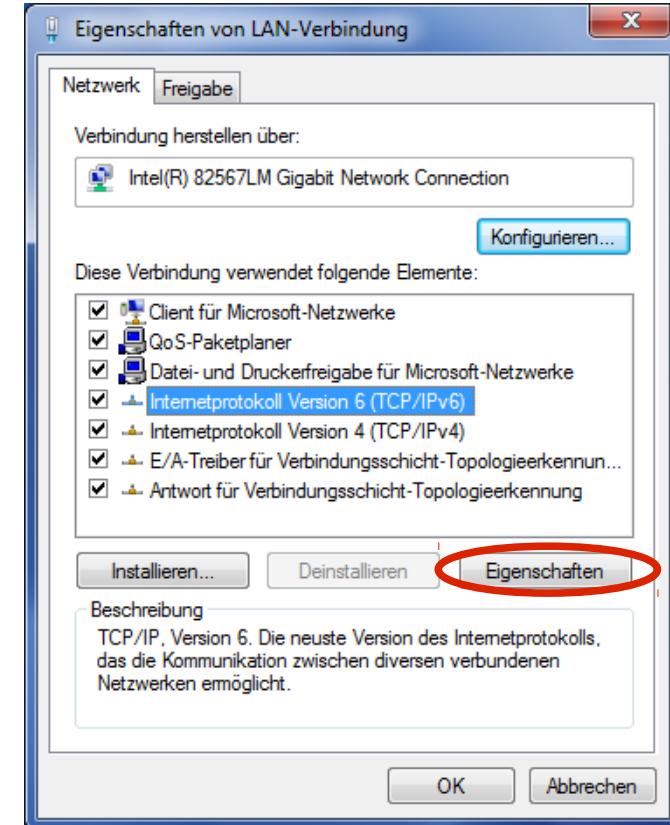
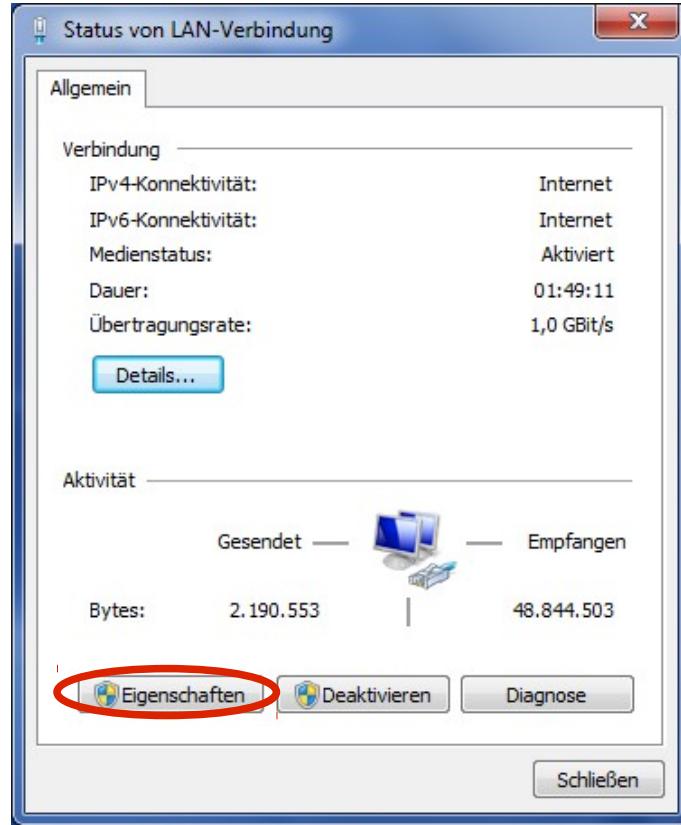


# Windows – Konfiguration

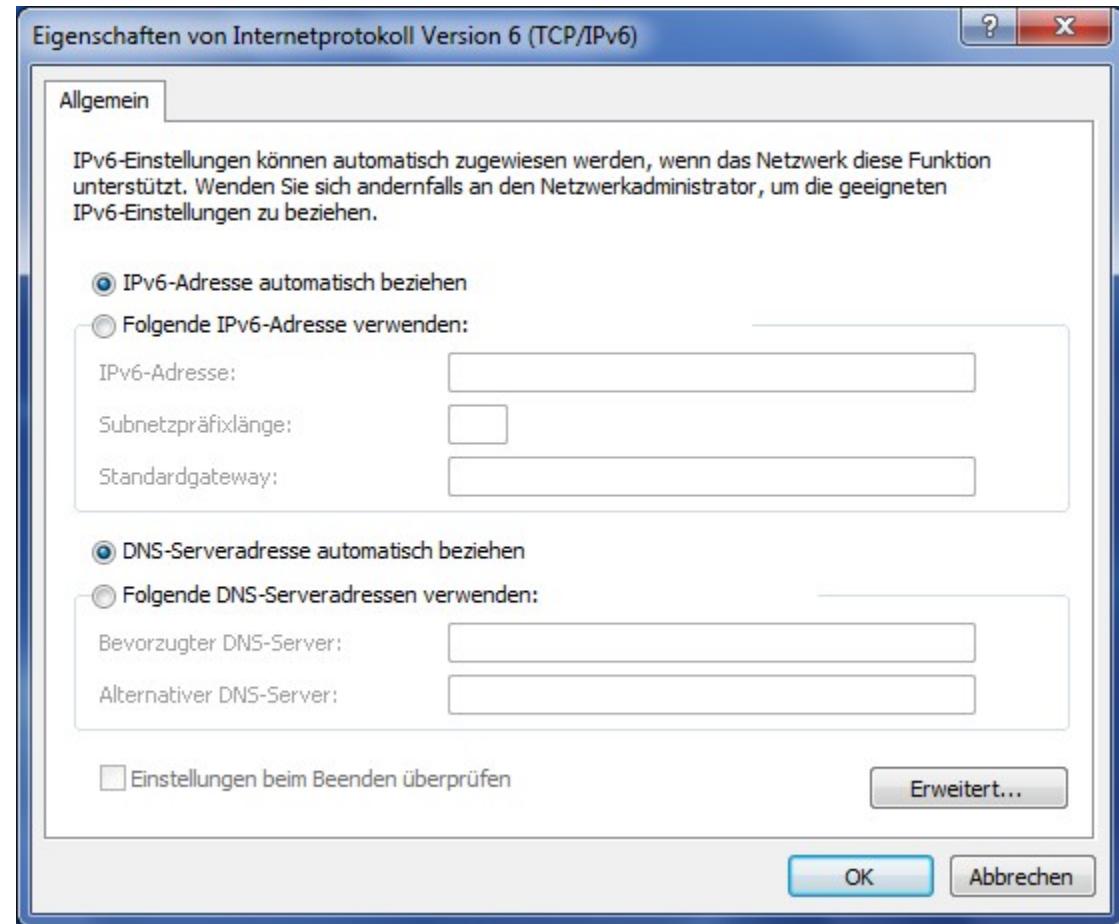


Eigenschaft	Wert
IPv4-Adresse	131.188.78.174
IPv4-Subnetzmaske	255.255.254.0
Lease erhalten	Mittwoch, 5. Dezember 2012 07:02:53
Lease läuft ab	Mittwoch, 5. Dezember 2012 21:44:13
IPv4-Standardgateway	131.188.79.1
IPv4-DHCP-Server	10.188.12.27
IPv4-DNS-Server	131.188.0.10 131.188.0.11
IPv4-WINS-Server	
NetBIOS über TCPIP akt...	Ja
IPv6-Adresse	2001:638:a000:3501::83bc:4eae
Lease erhalten	Mittwoch, 5. Dezember 2012 09:51:23
Lease läuft ab	Mittwoch, 5. Dezember 2012 11:51:23
Verbindungslokale IPv6-...	fe80::d46:6ab6:c404fd02%10
IPv6-Standardgateway	fe80::1%10
IPv6-DNS-Server	

# Windows – Konfiguration



# Windows Konfiguration



# Windows – Kommandozeile

- Konfiguration via `netsh`
  - Unter allen neueren Windows-Versionen vorhanden
  - Ermöglicht die Einstellung aller Netzparameter (teilweise über die GUI nicht konfigurierbar!)
  - Tool ähnelt Konfigurationskommandozeilen von Netzwerkequipment (z.B. IOS von Cisco)
  - Flexibel sowohl als „Shell“ als auch direkt von der Windows-Kommandozeile bedienbar
  - Gute Hilfefunktion
- Zusätzlich sind unter Windows die Programme `ipconfig` und `route` vorhanden, die den UNIX-Programmen `ifconfig` und `route` ähneln

# Windows – Kommandozeile

> *ipconfig*

Ethernetadapter LAN-Verbindung:

```
Verbindungsspezifisches DNS-Suffix: rrze.uni-erlangen.de
IPv6-Adresse . . . . . : 2001:638:a000:3501::83bc:4eae
Verbindungslokale IPv6-Adresse . . : fe80::d46:6ab6:c404:fd02%10
IPv4-Adresse . . . . . : 131.188.78.174
Subnetzmaske . . . . . : 255.255.254.0
Standardgateway . . . . . : fe80::1%10
                                131.188.79.1
```

# Windows Kommandozeile

> *netsh interface ipv6 show address*

Schnittstelle 1: Loopback Pseudo-Interface 1

Adresstyp	DAD-Status	Gültigkeit	Bevorzugt	Adresse
Andere	Bevorzugt	infinite	infinite	::1

Schnittstelle 12: Drahtlosnetzwerkverbindung

Adresstyp	DAD-Status	Gültigkeit	Bevorzugt	Adresse
Andere	Verworfen	infinite	infinite	fe80::c09:98c6:cc1f:abde%12

Schnittstelle 10: LAN-Verbindung

Adresstyp	DAD-Status	Gültigkeit	Bevorzugt	Adresse
DHCP	Bevorzugt	1h46m21s	46m21s	2001:638:a000:3501::83bc:4eae
Andere	Bevorzugt	infinite	infinite	fe80::d46:6ab6:c404:fd02%10

# Windows – Diagnose-Tools

> *tracert www.heise.de*

Routenverfolgung zu `www.heise.de [2a02:2e0:3fe:100::7]` über maximal 30 Abschnitte:

1	2 ms	2 ms	1 ms	sitak.gate.uni-erlangen.de [2001:638:a000:3501::2]
2	1 ms	<1 ms	<1 ms	constellation.gate.uni-erlangen.de [2001:638:a000::3336:33]
3	3 ms	1 ms	<1 ms	yamato.gate.uni-erlangen.de [2001:638:a000::3133:31]
4	<1 ms	<1 ms	<1 ms	xr-erl1-te1-3.x-win.dfn.de [2001:638:c:a039::1]
5	1 ms	<1 ms	<1 ms	cr-erl1-te0-7-0-0.x-win.dfn.de [2001:638:c:c038::1]
6	11 ms	12 ms	11 ms	cr-tub1-te0-7-0-6.x-win.dfn.de [2001:638:c:c08f::2]
7	36 ms	16 ms	15 ms	te3-1.c302.f.de.plusline.net [2001:7f8::3012:0:2]
8	16 ms	16 ms	16 ms	te2-4.c102.f.de.plusline.net [2a02:2e0:10:1:c::2]
9	15 ms	15 ms	15 ms	te6-2.c13.f.de.plusline.net [2a02:2e0:1::22]
10	15 ms	17 ms	16 ms	www.heise.de [2a02:2e0:3fe:100::7]

Ablaufverfolgung beendet.

# Windows – Diagnose-Tools

```
> tracert -4 www.heise.de
```

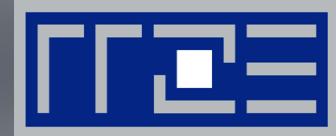
Routenverfolgung zu www.heise.de [193.99.144.85] über maximal 30 Abschnitte:

1	1 ms	<1 ms	1 ms	131.188.79.253
2	<1 ms	<1 ms	<1 ms	constellation.gate.uni-erlangen.de [131.188.20.201]
3	2 ms	<1 ms	1 ms	yamato.gate.uni-erlangen.de [131.188.20.106]
4	18 ms	5 ms	3 ms	xr-erl1-te1-3.x-win.dfn.de [188.1.234.229]
5	<1 ms	<1 ms	2 ms	cr-erl1-te0-0-0-0.x-win.dfn.de [188.1.145.113]
6	11 ms	10 ms	12 ms	cr-tub1-te0-7-0-6.x-win.dfn.de [188.1.145.234]
7	30 ms	26 ms	26 ms	te3-1.c101.f.de.plusline.net [80.81.192.132]
8	25 ms	25 ms	25 ms	heise2.f.de.plusline.net [82.98.98.106]
9	25 ms	26 ms	25 ms	www.heise.de [193.99.144.85]

Ablaufverfolgung beendet.



# DATENSCHUTZ UND PRIVATSPHÄRE



Eine weltweit eindeutige Adresse und damit eindeutig identifizierbar?

# Datenschutz und Privatsphäre

- Uni: Eindeutige/öffentliche Adresse mit IPv4 bereits üblich
- Typischer privater Anschluss: Netzwerkpräfix ähnelt bisheriger IPv4-Adresse, die üblicherweise bei Internetzugang verwendet wird (NAT) → Dynamische Adresspräfixe

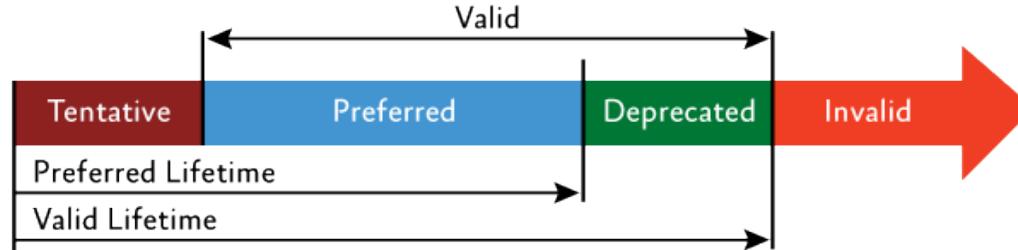
→ Bzgl Datenschutz sind Veränderungen geringer, als auf den ersten Blick meist vermutet

# Datenschutz und Privatsphäre

- IPv6 (mit statischen Adressen!) könnte sogar zur Erhöhung des Datenschutzes genutzt werden  
Siehe <http://heise.de/-1375692>
- Zukunft der Privatsphäre bei privaten Anschlüssen über Internet Service Provider (ISP) bleibt (nicht nur bzgl IPv6) ein spannendes Thema

# Privacy Extension

- Neues Problem: Host-Teil der IPv6-Adresse (Interface Identifier) enthält bei Autoconfiguration die MAC-Adresse  
→ Eindeutige Identifikation „überall“ möglich
- Verschärfung durch zunehmend „private“ Geräte (z.B. Smartphones)
- Lösung: Privacy Extensions (RFC 4941, ersetzt 3041)  
→ Temporäre, zufällige Adressen

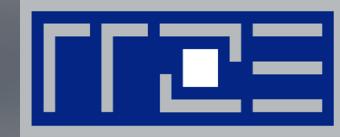


# Privacy Extension – Betriebssysteme

- Windows: Bei Desktop-Versionen ab XP vorhanden und standardmäßig aktiviert
  - Zusätzlich MAC per Default nicht in statischer IP-Adresse
  - <http://technet.microsoft.com/en-us/magazine/2007.08.cableguy.aspx>
  - ABER: Schwer „durch Netz“ deaktivierbar
- Linux: Im Kernel vorhanden, nicht per Default aktiviert
  - Aufgabe der jeweiligen Distribution
  - openSUSE: Ab 12.1 aktiviert
- Mac OS X: Ab 10.7 per Default aktiviert
- iOS: Ab 4.3 per Default aktiviert, nicht deaktivierbar!
  - In vorherigen Versionen nur nach Jailbreak aktivierbar
- Android: Nicht per Default aktiv, nur nach Jailbreak aktivierbar
- Siehe auch <http://heise.de/-1204783>



# GEFAHR DURCH DEN IPV6-TUNNEL TEREDO?



Teredo-Adressen, Verbindungsauflbau  
Nachteile und Gefahren

# Teredo

- Tunnel-Protokoll zur Migration von IPv4 nach IPv6
- Funktioniert hinter NAT-Routern, die nicht IPv6-fähig sind!
- Entwickelt von Microsoft, Standardisiert per RFCs
- In Windows enthalten, aber auch für andere OS verfügbar  
(Miredo <http://www.remlab.net/miredo/> )
- „Letzter Ausweg“ (MS), wenn sonst kein IPv6
- Siehe auch <http://heise.de/-221537> und  
<http://technet.microsoft.com/en-us/library/bb457011.aspx>

# Aufbau von Teredo-Adressen

- 32 Bit - Fester Präfix (2001::/32, also 2001:0000:.....)
- 32 Bit - IPv4-Adresse des Teredo-Servers
- 16 Bit - Flags
  - 4 Bit NAT-Typ, 12 Bit Zufallszahl zur Verbindungssicherung
- 16 Bit - Externe Port-Nummer des NAT-Routers
- 32 Bit - Externe IPv4-Adresse des NAT-Routers

# Teredo – Verbindungsaufbau

- Verbindungsaufbau zu Teredo-Server
  - Via IPv4-UDP-Port 3544
  - Automatische NAT-Erkennung – Wichtig!
  - Verbindung wird aufrecht erhalten (nötig bei NAT)
- Verbindungsaufbau zu IPv6-Ziel (vereinfachte Darstellung)
  - ICMPv6-Echo an Ziel (über *Teredo-Server*)
  - Antwort (Echo-Reply) des Ziels an zuständigen *Teredo-Relay* (ermittelt per IPv6-Routing, geographisch verteilt)
  - Relay sendet Antwort verpackt in IPv4-UPD-Paket an Client-Adresse (steht in *Teredo-Adresse*)
  - Letzter Schritt benötigt im Falle von *Restricted-NAT* eine *Bubble-to-open-Procedure* über den Teredo-Server
  - Anschließend Kommunikationen zu diesem Ziel direkt über Relay

# Teredo unter Windows

- Teredo-Interface taucht auf bei Ausgabe von  
> `netsh interface ipv6 show addresses`

- Weitere Informationen zu Teredo:

> `netsh interface ipv6 show teredo`

Teredo-Parameter

---

Typ	:	client
Servername	:	teredo.ipv6.microsoft.com.
Clientaktual.-intervall	:	30 Sekunden
Clientport	:	unspecified
Status	:	dormant

- Steht unter **Status** an Stelle von **dormant** oder **offline** der Wert **qualified**, ist der Tunnel aktiv!

# Teredo unter Windows

- Voreingestellter und ggf aktivierter Teredo-Server unter Windows
  - Bisher: teredo.ipv6.microsoft.com
  - Windows 8: win8.ipv6.microsoft.com
  - Ändern mittels

```
netsh interface ipv6 set teredo client teredo.example.com
```
- Deaktivieren auf dem Endsystem
  - `netsh interface ipv6 set teredo disable`
  - Deaktivieren von IPv6 als Ultima Ratio
- Deaktivieren an zentraler Stelle
  - UDP-Port 3544 auf Router oder Firewall sperren

# Teredo unter Windows

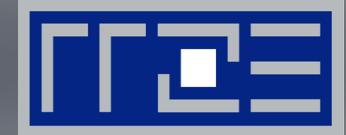
- Automatische Deaktivierung unter Windows ab Vista, wenn
  - das lokale Netz IPv6 spricht ODER
  - die Windows-Firewall abgeschaltet ist ODER
  - das LAN verwaltet wird (Active Directory)
- Besonderheit unter Windows ab Vista, wenn nur link-lokale oder Teredo-IPv6-Adressen
  - AAAA-Record nur ermittelt, wenn Anwendungsprogramm explizit danach fragt; sonst A-Record, wenn vorhanden  
→ Verkehr bevorzugt über IPv4
- Ungewisse Zukunft
  - Anscheinend wird Default-Server langfristig deaktiviert <http://heise.de/-1916499>
  - Aber Teredo für Xbox One nötig: <http://heise.de/-2060590>

# Teredo – Nachteile und Gefahren

- Performance
  - Viel Aufwand im Netz (Tunnel-Verbindungen auf NAT-Router)
  - Vergleichsweise ineffektiv
  - Nicht besonders stabil
- Umgeht "Schutz" durch NAT
  - Rechner weltweit erreichbar!
  - Aber: Windows ab Vista berücksichtigen IPv6 bei Firewall. Teredo automatisch deaktiviert, wenn Firewall deaktiviert.
- Normales Routing wird "umgangen"
- Verkehr wird über fremden Anbieter umgeleitet  
→ Problem mit Datenschutz!
- Siehe auch <http://heise.de/-270858>



# STABILITÄT UND SICHERHEIT



Probleme durch und mit IPv6  
im täglichen Einsatz

# Stabilität

- Dual-Stack (IPv6 und IPv4) schon oft gegeben
- Dank Link-Local oft schon IPv6 im lokalen Netz verwendet
- Klassisches Problem: IPv6 funktioniert nicht richtig
  - Relativ wenig Einschränkungen bemerkbar (Trägheit)
  - Zuordnung der Fehlerbilder erschwert
- Durchaus möglich: Kein DHCPv4, aber IPv6 SLAAC
  - Es funktioniert nur IPv6!
  - Zuordnung der Fehlerbilder ebenfalls erschwert
- IPv6-fähige Hosts können (versehentlich) IPv6- und sogar IPv4-Netzwerke massiv stören
  - *Rogue Router Advertisements*: Windows mit Internet-Connection-Sharing und 6to4-Tunnel werden zu IPv6-Router und „bieten ihre Dienste an“

# Stabilität

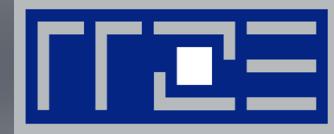
- IPv6 immer im Hinterkopf behalten, wenn Probleme auftauchen!
- Test mit Browsern möglich:  
<http://test-ipv6.com/>  
<http://www.heise.de/netze/tools/ip/>  
<http://www.bieringer.de/>

# Sicherheit

- IPv6 ist relativ neu → Viele **Bugs** lauern da draußen!  
Auch „alte Bekannte“ wie der *Ping of Death* aus Windows 95:  
<https://technet.microsoft.com/library/security/ms13-065>
- Server statische, Clients größtenteils dynamische Adressen. Wie bisher.  
IPv6 bringt „nur“ mehr Flexibilität!
- Probleme mit Firewalls
  - Dynamische Adressvergabe
  - Mehrere IP-Adressen pro InterfaceAber: Firewalls heute oft „unabhängig“ von der IP-Adresse  
Dennoch: Feature- und Performance-Probleme mit IPv6
- Sonstige Sicherheitsaspekte meist weder besser noch schlechter als bei IPv4 (<https://www.thc.org/thc-ipv6/>)

# IPv6 an der FAU

- Wesentliches bereits in Teil 1 zu finden
- Weitere Informationsquellen
  - Webseiten des RRZE  
<http://www.rrze.fau.de/infrastruktur/kommunikationsnetz/ipv6.shtml>
  - Benutzerinformationen (BI) 89 (11/2013), S. 23ff



Vielen Dank!