

# Praktikum Rechnernetze

Protokoll zu Versuch 4 (IPv6) von Gruppe 1

---

Jakob Waibel   Daniel Hiller   Elia Wüstner   Felix Pojtinger

2021-11-09

# Einführung

---

Diese Materialien basieren auf Professor Kiefers “Praktikum Rechnernetze”-Vorlesung der HdM Stuttgart.

**Sie haben einen Fehler gefunden oder haben einen Verbesserungsvorschlag?** Bitte eröffnen Sie ein Issue auf GitHub ([github.com/pojntfx/uni-netpractice-notes](https://github.com/pojntfx/uni-netpractice-notes)):



**Abbildung 1:** QR-Code zum Quelltext auf GitHub

Dieses Dokument und der enthaltene Quelltext ist freie Kultur bzw. freie Software.



**Abbildung 2:** Badge der AGPL-3.0-Lizenz

Uni Network Practice Notes (c) 2021 Jakob Waibel, Daniel Hiller,  
Elia Wüstner, Felix Pojtinger

SPDX-License-Identifier: AGPL-3.0

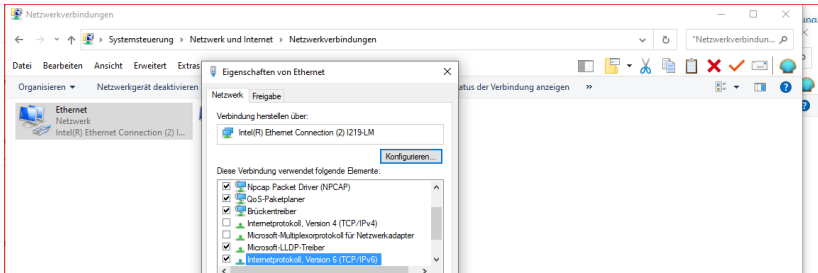
# IPv6-Adressen

---

# IPv6-Adressen

## Voreinstellung für die Aufgaben - deaktivieren von IPv4 und aktivieren von IPv6 unter Windows.

Um IPv4 zu deaktivieren und IPv6 zu aktivieren, muss man in den Netzwerkeinstellungen zum jeweiligen Adapter über den Pfad Systemsteuerung > Netzwerk und Internet > Netzwerkverbindungen > A navigieren. Hier wurde der Haken bei IPv6 (Internetprotokoll, Version6) gesetzt und bei IPv4 (Internetprotokoll, Version4) entfernt.



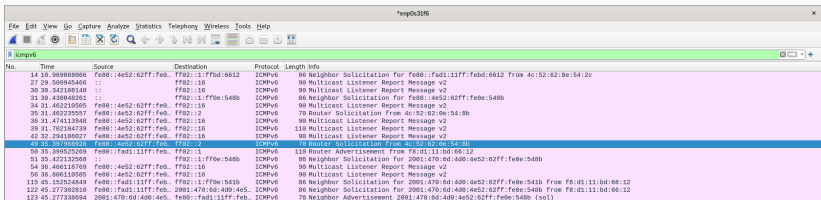
# IPv6 und DNS

---

Identifizieren Sie mit Wireshark die Pakete mit denen der Router im Netz das Prefix mitteilt. Welches Protokoll wird dafür benutzt und um welchen Type handelt es sich und wie lautet die Zieladresse des Pakets?

Das verwendete Protokoll ist wie auch in den unten stehenden Screenshots zu sehen ICMPv6. Die Types sind Router Solicitation und Router Advertisement. Die Zieladresse des Pakets ist die Multicast-Adresse ff02::1.

Router Solicitation:



\*enp0s31f6

No.	Time	Source	Destination	Protocol	Length	Info
14	10.369086066	fe80::4e52:62ff:feb::ff02::1:ffbd:6612	ff02::1	ICMPv6	86	Neighbor Solicitation For fe80::fadi:11ff:febd:6612 from 4c:52:62:be:54:2c
27	29.509945466	::	ff02::1	ICMPv6	98	Multicast Listener Report Message v2
30	30.342180148	::	ff02::1	ICMPv6	98	Multicast Listener Report Message v2
31	30.438948593	::	ff02::1:ff0e:548b	ICMPv6	86	Neighbor Solicitation For fe80::4e52:62ff:febe:548b
34	31.462210505	fe80::4e52:62ff:feb::ff02::1	ff02::1	ICMPv6	98	Multicast Listener Report Message v2
35	31.462235557	fe80::4e52:62ff:feb::ff02::1	ff02::1	ICMPv6	78	Router Solicitation from 4c:52:62:0e:54:8b
36	31.474113948	fe80::4e52:62ff:feb::ff02::1	ff02::1	ICMPv6	98	Multicast Listener Report Message v2
39	31.782184739	fe80::4e52:62ff:feb::ff02::1	ff02::1	ICMPv6	118	Multicast Listener Report Message v2
42	32.294188027	fe80::4e52:62ff:feb::ff02::1	ff02::1	ICMPv6	98	Multicast Listener Report Message v2
73	90.330105720	fe80::4e52:62ff:feb::ff02::1	ff02::1	ICMPv6	118	Router Advertisement from fe80::fadi:11ff:febd:6612
90	95.399025269	fe80::fadi:11ff:feb::ff02::1	ff02::1	ICMPv6	118	Router Advertisement from fe80::fadi:11ff:febd:6612
91	95.422132568	::	ff02::1:ff0e:548b	ICMPv6	86	Neighbor Solicitation For 2001:470:6d:400:4e52:62ff:febe:548b
94	96.446118789	fe80::4e52:62ff:feb::ff02::1	ff02::1	ICMPv6	98	Multicast Listener Report Message v2
96	96.806110585	fe80::4e52:62ff:feb::ff02::1	ff02::1	ICMPv6	98	Multicast Listener Report Message v2
115	95.325248449	fe80::fadi:11ff:feb::ff02::1:ff0e:548b	ff02::1	ICMPv6	86	Neighbor Solicitation For 2001:470:6d:400:4e52:62ff:febe:548b from fe80::fadi:11ff:febd:6612
122	95.477380518	fe80::fadi:11ff:feb::2001:470:6d:400:4e52:62ff:febe:548b	2001:470:6d:400:4e52:62ff:febe:548b	ICMPv6	86	Neighbor Solicitation For 2001:470:6d:400:4e52:62ff:febe:548b from fe80::fadi:11ff:febd:6612
123	95.277388094	2001:470:6d:400:4e52:62ff:febe:548b	fe80::fadi:11ff:feb::ff02::1	ICMPv6	78	Neighbor Advertisement 2001:470:6d:400:4e52:62ff:febe:548b (sol)

Frame 40: 78 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface enp0s31f6, id 0



# Neighbor Solicitation

---

# Neighbor Solicitation

Starten Sie den „Kabelhai“ und pingen Sie ihren Nachbarrechner. Welches Protokoll/Type wird anstatt ARP zur Ermittlung der MAC-Adressen verwendet?

*Windows*

The screenshot shows a Windows command prompt window on the left and a Wireshark network traffic analysis window on the right.

**Command Prompt:**

```
PS C:\WINDOWS\system32> netsh int ipv6 del neigh  
OK  
PS C:\WINDOWS\system32> ping -6 2001:470:6d:4d0:4e52:62ff:fe0e:548b  
Ping wird ausgeführt für 2001:470:6d:4d0:4e52:62ff:fe0e:548b mit 32 Bytes Daten:  
Antwort von 2001:470:6d:4d0:4e52:62ff:fe0e:548b: Zeit=1ms  
Antwort von 2001:470:6d:4d0:4e52:62ff:fe0e:548b: Zeit=1ms  
Antwort von 2001:470:6d:4d0:4e52:62ff:fe0e:548b: Zeit=1ms  
Antwort von 2001:470:6d:4d0:4e52:62ff:fe0e:548b: Zeit=1ms  
Ping-Statistik für 2001:470:6d:4d0:4e52:62ff:fe0e:548b:  
Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0  
(0% Verlust)  
ca. Zeitangaben in Millisek.:  
Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms  
PS C:\WINDOWS\system32>
```

**Wireshark:**

The Wireshark window shows a packet capture on the Ethernet interface (2) 019-10A Ethernet. The packet list shows several Neighbor Solicitation (Type 135) and Neighbor Advertisement (Type 136) packets. The packet details pane shows the structure of a Neighbor Solicitation packet, including the destination MAC address (ff02::1:3) and the destination IPv6 address (2001:470:6d:4d0:4e52:62ff:fe0e:548b).

**Abbildung 16:** Solicitation und Advertisement-Pakete in Wireshark - Windows

*Linux*

```
$ sudo ip neigh flush dev enp0s31f6
```

# IPv6-Header

---

# IPv6-Header

Starten Sie Wireshark und senden sie ein ping an einen IPv6-fähigen Webserver ([www.ix.de](http://www.ix.de), <http://www.heise.de>, <http://www.kame.net>), stoppen Sie Wireshark und schauen sich den Trace an.

## Windows

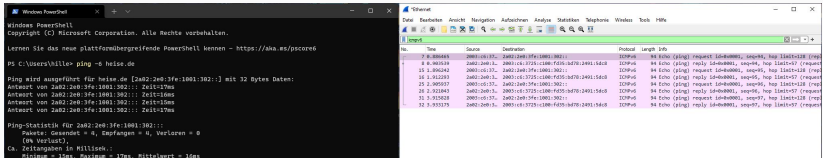


Abbildung 18: Ping Heise

## Linux

```
$ ping www.kame.net
```

```
PING www.kame.net (2001:2f0:0:8800:226:2dff:fe0b:431)
```

# Privacy Extension

---

# Privacy Extension

**Tragen Sie weitere Informationen zur „Privacy Extension“ (vor allem auch zur Konfiguration unter Windows und Ubuntu) zusammen und versuchen hier im Versuch die Einstellungen für die „Privacy Extension“ auf beiden Rechnern (Windows und Ubuntu) zu realisieren.**

Privacy Extensions sind dafür da, Rückschluss auf Nutzer:innen schwerer zu machen, indem der Hostanteil der IPv6-Adressen anonymisiert wird. Privacy Extensions entkoppeln Interface Identifier und MAC-Adresse und erzeugen diese nahezu zufällig. Mit diesen periodisch wechselnden Adressen werden dann ausgehende Verbindungen hergestellt, was den Rückschluss auf *einzelne* Nutzer:innen erschwert. Mit Hilfe der Privacy Extensions kann man also nicht mehr einzelne Nutzer:innen identifizieren. Was allerdings trotzdem möglich ist, ist das Identifizieren über den Präfix, welcher allerdings nur Informationen zum Netzwerk bereitstellt. Wenn der

# Feste IPv6-Adressen

---

# Feste IPv6-Adressen

Weisen Sie in dieser Aufgabe ihrem Netzwerkinterface eine feste sinnvolle (heißt: Der Prefix ist weiterhin gültig) IPv6-Adresse zu.

*Windows*

Eigenschaften von Internetprotokoll, Version 6 (TCP/IPv6) ✕

**Allgemein**

IPv6-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IPv6-Einstellungen zu beziehen.

☐ IPv6-Adresse automatisch beziehen

☒ Folgende IPv6-Adresse verwenden:

IPv6-Adresse:

Subnetzpräfixlänge:

Standardgateway:

☐ DNS-Serveradresse automatisch beziehen

☒ Folgende DNS-Serveradressen verwenden:

Bevorzugter DNS-Server:



# Lease-Zeiten

---

Die Werte für “Maximale bevorzugte Gültigkeitsdauer” und “Maximale Gültigkeitsdauer” setzt man in Windows über die Schlüssel `maxpreferredlifetime` und `maxvalidlifetime`, die Zeitangaben in Tagen (d), Stunden (h), Minuten (m) und Sekunden (s) entgegennehmen. Wie sind diese Parameter bei Ihnen gesetzt?

*Windows*

```
netsh interface ipv6 show privacy
```

```
Parameter für temporäre Adressen
```

```
-----  
Temporäre Adresse verwenden           : enabled  
Versuch, doppelte Adr. zu entdecken  : 3  
Maximale Gültigkeitsdauer             : 7d  
Maximale bevorzugte Gültigkeitsdauer: 7d  
Regenerationszeit                     : 5s  
Maximale Verzögerungszeit             : 10m
```

# OS-Updates

---

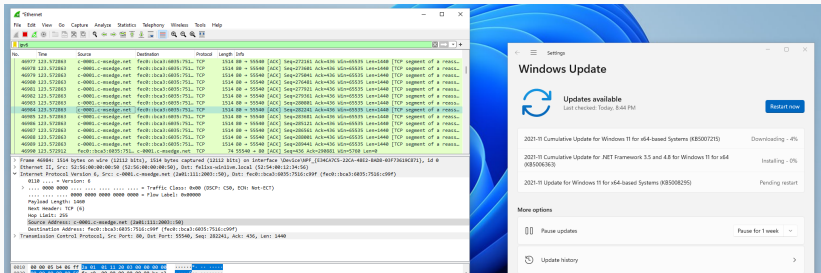
## Lässt sich eigentlich Windows über IPv6 updaten? Was sagt Wireshark dazu?

### Windows

Unter Windows wurde das Update ohne Probleme installiert.

Windows Update verfügt über vollen IPv6-Support.

(<https://serverfault.com/questions/844107/windows-server-update-on-ipv6-only-network>). Dies konnte auch mittels Wireshark validiert werden:



The image displays two side-by-side screenshots. The left screenshot shows a Wireshark packet capture of an IPv6 update process. The packet list table includes the following entries:

No.	Time	Source	Destination	Protocol	Length	Info
40077	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40078	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40079	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40080	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40081	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40082	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40083	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40084	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40085	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40086	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40087	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40088	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40089	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0
40090	123.572843	c-0001.c-madag.net	fe8b::c0a3:0001:751c::	TCP	1514	80 → 55540 [ACK] Seq=2722431 Ack=436 Win=65535 Len=0

The packet details pane shows the following information:

- Frame 40084: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on Interface UserNIC6, [Ethernet II, Src: c-0001.c-madag.net (2a01:1111:2003:150), Dst: fe8b::c0a3:0001:751c::c09f] (fe8b::c0a3:0001:751c::c09f)
- Ethernet II, Src: c-0001.c-madag.net (2a01:1111:2003:150), Dst: fe8b::c0a3:0001:751c::c09f
- Internet Protocol Version 6, Src: c-0001.c-madag.net (2a01:1111:2003:150), Dst: fe8b::c0a3:0001:751c::c09f
- TCP, Src Port: 80, Dst Port: 55540, Seq: 2722431, Win: 65535, Len: 0

The right screenshot shows the Windows Update settings window. It displays the following information:

- Windows Update
- Updates available
- Last checked: Today, 8:44 PM
- Restart now
- 2021-11 Cumulative Update for Windows 11 for x64-based Systems (KB5007215) Downloading - 4%
- 2021-11 Cumulative Update for .NET Framework 3.5 and 4.8 for Windows 11 for x64 (KB5006363) Installing - 0%
- 2021-11 Update for Windows 11 for x64-based Systems (KB5006295) Pending restart
- More options
- Pause updates
- Update history