

# **Praktikum Rechnernetze**

Protokoll zu Versuch 4 (IPv6) von Gruppe 1

---

Jakob Waibel    Daniel Hiller    Elia Wüstner    Felix Pojtinger

2021-11-09

# Einführung

---

# Mitwirken

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)):



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

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



**Figure 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-Addressen

---

# IPv6-Addressen

## 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

---

## 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 Protkoll 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:

*exp0\$31f6							
File	Edit	View	Capture	Analyze	Statistics	Telephone	Wireless
No.	Time	Source	Destination	Protocol	Length	Info	
14	20:59:08.869686	Feb0: :e452:6fff:feb..	ff02::1:ffbd:6612	ICMPv6	88	Neighbor Solicitation for fe80::fadi:1fff:febd:6612 from 4c:52:62:8e:54:2c	
27	20:59:09.451466	...	ff02::1:ffbd:6612	ICMPv6	88	Multicast Listener Report Message v2	
30	34:21:00.181140	...	ff02::1:ffff:16	ICMPv6	88	Multicast Listener Report Message v2	
31	30.43848848261	...	ff02:1:ffff:54bb	ICMPv6	88	Neighbor Solicitation for fe80:::e452:6fff:febe:54bb	
31	30.43848848261	...	ff02:1:ffff:54bb	ICMPv6	96	Multicast Listener Report Message v2	
33	31.4622105965	Feb0: :e452:6fff:feb..	ff02::1:ffff:16	ICMPv6	96	Neighbor Solicitation for fe80:::e452:6fff:febe:54:8b	
36	31.4745139348	Feb0: :e452:6fff:feb..	ff02::1:ffff:16	ICMPv6	96	Neighbor Solicitation for fe80:::e452:6fff:febe:54:8b	
39	31.7921847349	Feb0: :e452:6fff:feb..	ff02::1:ffff:16	ICMPv6	96	Neighbor Solicitation for fe80:::e452:6fff:febe:54:8b	
42	32.294189027	Feb0: :e452:6fff:feb..	ff02::1:ffff:16	ICMPv6	96	Neighbor Solicitation for fe80:::e452:6fff:febe:54:8b	
43	32.294189027	Feb0: :e452:6fff:feb..	ff02::1:ffff:16	ICMPv6	96	Neighbor Solicitation for fe80:::e452:6fff:febe:54:8b	
50	35.399525269	Feb0: fadi:1fff:feb..	ff02::1:ffff:54:10	ICMPv6	118	Router Advertisement From fe80::d1:11ff:hd:66:12	
51	35.422132262	...	ff02:1:ffff:54:10	ICMPv6	88	Neighbor Solicitation for 2001::470::6d:480::e452:6fff:febe:54:8b	
54	35.4661219769	Feb0: :e452:6fff:feb..	ff02::1:ffff:16	ICMPv6	96	Multicast Listener Report Message v2	
56	36.3861190823	Feb0: :e452:6fff:feb..	ff02::1:ffff:16	ICMPv6	96	Multicast Listener Report Message v2	
113	45.1525248489	Feb0: fadi:1fff:feb..	ff02:1:ffff:54:10	ICMPv6	88	Neighbor Solicitation for 2001::470::6d:480::e452:6fff:febe:54:10	
122	45.2773982838	2001::470::6d:480::4e5..	ff02:1:ffff:54:10	ICMPv6	88	Neighbor Solicitation for 2001::470::6d:480::e452:6fff:febe:54:10	
123	45.2773982838	2001::470::6d:480::4e5..	ff00: fadi:1fff:feb..	ICMPv6	768	Router Advertisement 2001::470::6d:480::e452:6fff:febe:54:10 (sol)	

## **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



Figure 16: Solicitation und Advertisement-Pakete in Wireshark – Windows

Linux

```
$ sudo ip neigh flush dev enp0s31f6
$ ping6 fe80::fad1:11ff:febd:6612
```

## IPv6-Header

---

# IPv6-Header

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

Windows



Figure 18: Ping Heise

Linux

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

```
PING www.kame.net(2001:2f0:0:8800:226:2dff:fe0b:4311)  
64 bytes from 2001:2f0:0:8800:226:2dff:fe0b:4311 (2
```

## 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, Rückchluss 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

## Feste IPv6-Addressen

---

# Feste IPv6-Addressen

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

Windows



## **Lease-Zeiten**

---

## 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 : 10s
```

## OS-Updates

---

# 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:

