Praktikum Rechnernetze

Versuch 3

vom 12.5.2020

IPv6

Protokoll Gruppe 1

Yannick Möller (ym018)

Bernd Maier (bm075)

Michael Vanhee (mv068)

Rebecca Mombrei (rm048)

Aufgabe 1

Mithilfe von **netsh int ipv6 show addr** lassen sich alle den Adaptern zugeordneten IP-Adressen anzeigen:

```
C:\WINDOWS\system32>netsh int ipv6 show addr

Schnittstelle 11: VirtualBox Host-Only Network #3

Adresstyp  DAD-Status  Gültigkeit Bevorzugt Adresse
---------  ----------- ---------- ---------- ------------------------
Andere     Bevorzugt     infinite   infinite fe80::800:27ff:fe00:b%11

Schnittstelle 1: Loopback Pseudo-Interface 1

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

Schnittstelle 7: Ethernet 2

Adresstyp  DAD-Status  Gültigkeit Bevorzugt Adresse
---------  ----------- ---------- ---------- ------------------------
Öffentlich Bevorzugt    23h58m21s   3h58m21s 2001:470:26:5f7:4e52:62ff:fe0e:5401
Andere     Bevorzugt     infinite   infinite fe80::4e52:62ff:fe0e:5401%7


C:\WINDOWS\system32>_
```

MIttels **nmap -sP 141.62.66.0/24** kann man einen Ping Scan machen. Er gibt ausschließlich Auskunft darüber, ob die Hosts in dem Netz online sind.

```
Nmap done: 256 IP addresses (46 hosts up) scanned in 11.13 seconds
```

Mit **ip -6 addr** kann man die zugewiesene IP am eth0 (Achtung, neues Bezeichnungsschema seit kurzem bei Linux) erkennen.

```
File  Edit  View  Search  Terminal  Help
root@is02:~# ip -6 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1000
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 2001:470:26:5f7:4e52:62ff:fe0e:5441/64 scope global dynamic mngtmpaddr
       valid_lft 86375sec preferred_lft 14375sec
    inet6 fe80::4e52:62ff:fe0e:5441/64 scope link
       valid_lft forever preferred_lft forever
root@is02:~#
```

Woraus setzt sich die Link-Lokale-Adresse zusammen und erkennen Sie das EUI-64 Format?

Die Link-Local Adresse setzt sich aus zwei Teilen zusammen:

Präfix	Suffix
Beginnt mit fe80 und wird mit Nullen aufgefüllt.	Der Suffix besteht aus der MAC-Adresse des Netzwerk-Interface, welche in das EUI-64 Format umgewandelt wurde. Dieses erkennt man daran, dass die MAC-Adresse in der Mitte aufgespalten wird und mithilfe der Füllbytes ff:fe auf 64 Bit erweitert wird.
fe80:0000:0000:0000 (mehrere gleiche Blöcke kann man mit :: abkürzen => fe80::)	**4e52:62ff:fe0e:5401**

Wie lautet der Prefix und die Host-ID der Global-Unicast-Adresse?

Der Prefix der Global-Unicast Adresse lautet bei meinem Laborrechner 2 (2001:470:26:5f7). Die Host-ID entspricht der meines Rechners (siehe Tabelle Suffix oben).
Testen Sie die Netzwerkverbindung zwischen dem Linux und dem Windows-Rechner mit einem Ping (IPv6)?
Der Ping zum benachbarten Linux-Rechner läuft erfolgreich ab:

```
C:\WINDOWS\system32>ping -6 fe80::4e52:62ff:fe0e:54de

Ping wird ausgeführt für fe80::4e52:62ff:fe0e:54de mit 32 Bytes Daten:
Antwort von fe80::4e52:62ff:fe0e:54de: Zeit<1ms
Antwort von fe80::4e52:62ff:fe0e:54de: Zeit<1ms
Antwort von fe80::4e52:62ff:fe0e:54de: Zeit<1ms
Antwort von fe80::4e52:62ff:fe0e:54de: Zeit<1ms

Ping-Statistik für fe80::4e52:62ff:fe0e:54de:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

Windows <> Windows

```
C:\Windows\System32>ping -6 fe80::4e52:62ff:fe0e:5401

Ping wird ausgeführt für fe80::4e52:62ff:fe0e:5401 mit 32 Bytes Daten:
Zielhost nicht erreichbar.
Antwort von fe80::4e52:62ff:fe0e:5401: Zeit=1ms
Antwort von fe80::4e52:62ff:fe0e:5401: Zeit<1ms
Antwort von fe80::4e52:62ff:fe0e:5401: Zeit<1ms

Ping-Statistik für fe80::4e52:62ff:fe0e:5401:
    Pakete: Gesendet = 4, Empfangen = 3, Verloren = 1
    (25% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms
```

Lassen Sie sich die Routen anzeigen und ermitteln Sie die „Default Route“.

Windows:
Die Routen lassen sich mithilfe des Befehls **netsh interface ipv6 show route** anzeigen.

```
C:\WINDOWS\system32>netsh interface ipv6 show route

Veröff.  Typ       Met   Präfix                    Idx  Gateway/Schnittstelle
-------  --------  ----  ------------------------  ---  --------------------
Nein     Manuell   256  ::/0                       7  fe80::fad1:11ff:fea9:ee84
Nein     System    256  ::1/128                    1  Loopback Pseudo-Interface 1
Nein     Manuell   256  2001:470:26:5f7::/64       7  Ethernet 2
Nein     System    256  2001:470:26:5f7:4e52:62ff:fe0e:5401/128    7  Ethernet 2
Nein     System    256  fe80::/64                 11  VirtualBox Host-Only Network #3
Nein     System    256  fe80::/64                  7  Ethernet 2
Nein     System    256  fe80::800:27ff:fe00:b/128  11  VirtualBox Host-Only Network #3
Nein     System    256  fe80::4e52:62ff:fe0e:5401/128    7  Ethernet 2
Nein     System    256  ff00::/8                   1  Loopback Pseudo-Interface 1
Nein     System    256  ff00::/8                  11  VirtualBox Host-Only Network #3
Nein     System    256  ff00::/8                   7  Ethernet 2


C:\WINDOWS\system32>
```

Die Default Route ist ::/0.

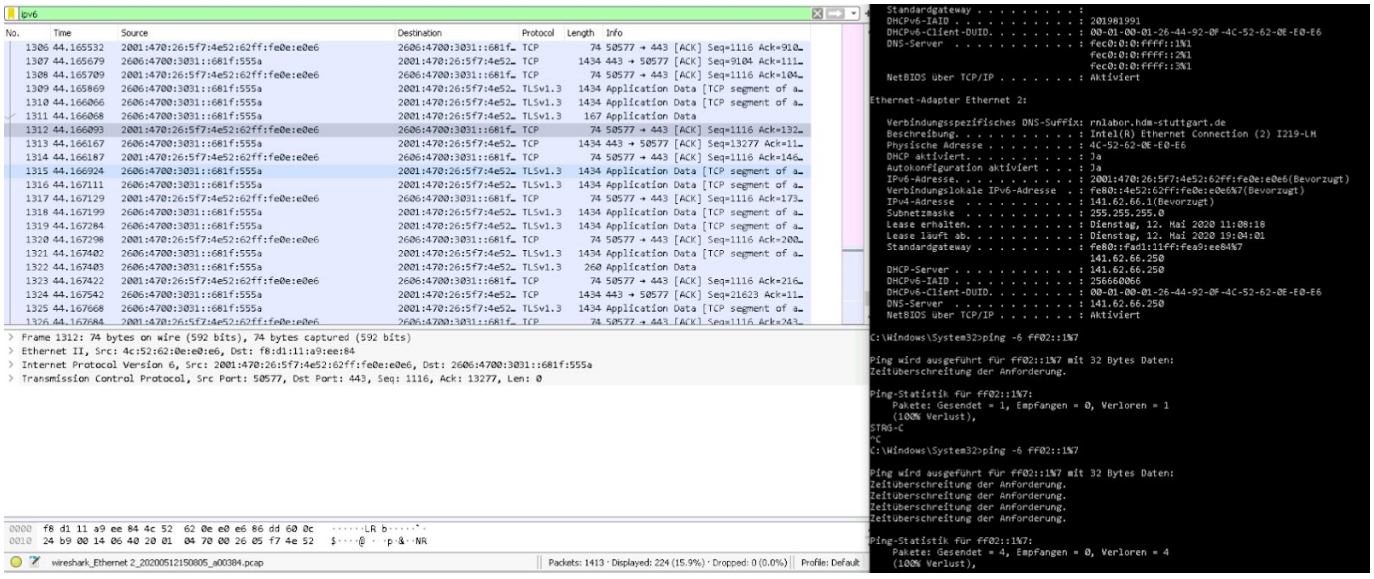
Linux:

```
root@is01:/home/praktikum# netstat -rn -A inet6
Kernel IPv6 routing table
Destination                    Next Hop                   Flag Met Ref Use If
2001:470:26:5f7::/64           ::                         UAe  256 1     0 enp0s31f6
fe80::/64                      ::                         U    256 3     0 enp0s31f6
::/0                           fe80::fad1:11ff:fea9:ee84  UGDAe 1024 1     0 enp0s31f6
::1/128                        ::                         Un   0   4     0 lo
2001:470:26:5f7:4e52:62ff:fe0e:54de/128 ::                          Un   0   2     0 enp0s31f6
fe80::4e52:62ff:fe0e:54de/128  ::                         Un   0   6     0 enp0s31f6
ff00::/8                       ::                         U    256 6     0 enp0s31f6
::/0                           ::                         !n   -1  1     0 lo
```

Aufgabe 2

Wer antwortet auf ping -6 ff02::1%<Schnittstellennummer> und ping -6 ff02::2%<Schnittstellennummer> mit welchem Protokoll und warum?

via Guacamole von 141.62.66.1 mit **ping -6 ff02::1%7**



(für Linux)
ping6 ff02::1%2 (enp0s31f6)

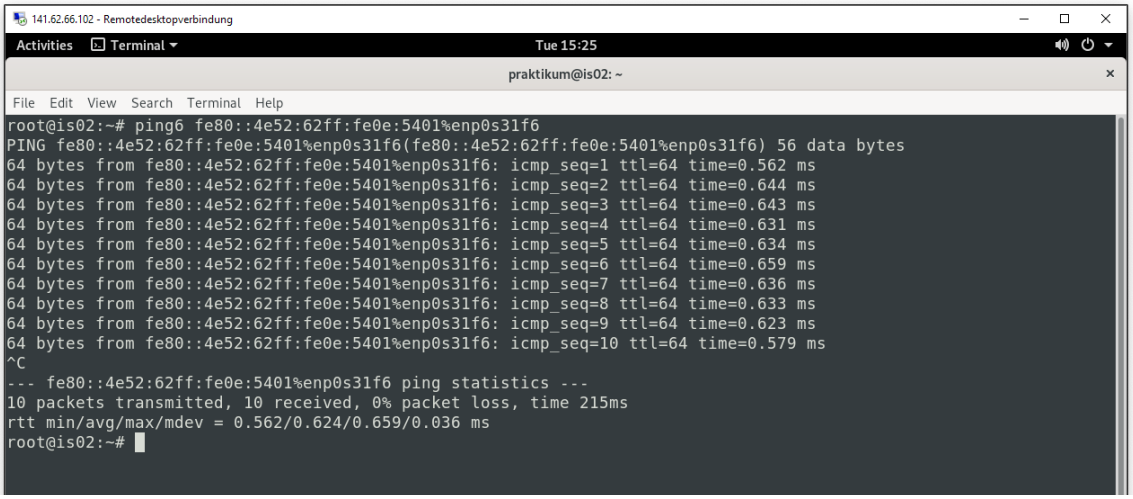
```
root@is01:/home/praktikum# ping -6 ff02::1%2
PING ff02::1%2(ff02::1%enp0s31f6) 56 data bytes
64 bytes from fe80::4e52:62ff:fe0e:54de%enp0s31f6: icmp_seq=1 ttl=64 time=0.055 ms
64 bytes from fe80::4e52:62ff:fe0e:5441%enp0s31f6: icmp_seq=1 ttl=64 time=0.429 ms (DUP!)
64 bytes from fe80::219:99ff:fef9:7ec2%enp0s31f6: icmp_seq=1 ttl=64 time=0.546 ms (DUP!)
64 bytes from fe80::219:99ff:fef9:7c8a%enp0s31f6: icmp_seq=1 ttl=64 time=0.600 ms (DUP!)
64 bytes from fe80::a4e2:e2ff:fecd:e47d%enp0s31f6: icmp_seq=1 ttl=64 time=0.608 ms (DUP!)
64 bytes from fe80::b858:f6ff:fe60:f766%enp0s31f6: icmp_seq=1 ttl=64 time=0.616 ms (DUP!)
64 bytes from fe80::dcab:6dff:fef8:ad58%enp0s31f6: icmp_seq=1 ttl=64 time=0.621 ms (DUP!)
64 bytes from fe80::4e52:62ff:fe0e:53e6%enp0s31f6: icmp_seq=1 ttl=64 time=0.625 ms (DUP!)
```

ping6 ff02::2%2 (enp0s31f6)

```
praktikum@is01:~$ ping6 ff02::2%2
PING ff02::2%2(ff02::2%enp0s31f6) 56 data bytes
64 bytes from fe80::fad1:11ff:fea9:ee84%enp0s31f6: icmp_seq=1 ttl=64 time=0.622 ms
64 bytes from fe80::fad1:11ff:fea9:ee84%enp0s31f6: icmp_seq=2 ttl=64 time=0.390 ms
64 bytes from fe80::fad1:11ff:fea9:ee84%enp0s31f6: icmp_seq=3 ttl=64 time=0.364 ms
64 bytes from fe80::fad1:11ff:fea9:ee84%enp0s31f6: icmp_seq=4 ttl=64 time=0.279 ms
64 bytes from fe80::fad1:11ff:fea9:ee84%enp0s31f6: icmp_seq=5 ttl=64 time=0.358 ms
64 bytes from fe80::fad1:11ff:fea9:ee84%enp0s31f6: icmp_seq=6 ttl=64 time=0.390 ms
```

Die Multicast-Adresse ff02::2 adressiert alle Router im Netz, ff02::1 alle Interfaces (also Router und Endgeräte), die IPv6-fähig sind.

Der Michi hat die Becca angepingt.



ICMPv6 und IPv6 Protokoll

Es wird das ICMPv6 Protokoll verwendet, da es zusammen mit dem IPv6 Protokoll heraus gekommen ist. ICMPv6 ist für IPv6 zwingend erforderlich, nicht so wie bei ICMP und IPv4. Ein blockieren von ICMPv6 ist nicht möglich, denn dann funktioniert IPv6 schlichtweg nicht mehr.

Können Sie einzelne Nodes anhand der MAC-Adresse (siehe Anhang) identifizieren?

Von den folgenden Stationen kam eine Antwort auf den ping an **ping -6 ff02::1%7**:

No.	Time	Source	Destination	Protocol	Length	Info
305	4.258708	fe80::219:99ff:fef9:7ec2	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
306	4.258708	fe80::a4e2:e2ff:fecd:e47d	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
307	4.258708	fe80::b858:f6ff:fe60:f766	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
308	4.258708	fe80::dcab:6dff:fef8:ad58	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
309	4.258709	fe80::4e52:62ff:fe0e:5441	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
312	4.258715	fe80::4e52:62ff:fe0e:541b	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
313	4.258715	fe80::4e52:62ff:fe0e:53f1	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
314	4.258715	fe80::4e52:62ff:fe0e:53d0	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
315	4.258715	fe80::4e52:62ff:fe0e:54de	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
316	4.258716	fe80::4e52:62ff:fe0e:e0e3	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
318	4.258716	fe80::4e52:62ff:fe0e:53e6	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
328	4.258892	fe80::219:99ff:fef9:7c8a	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
331	4.258971	fe80::b04f:d6ff:fe65:93c7	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
332	4.258971	fe80::e864:1cff:fee0:c82f	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
333	4.258971	fe80::fad1:11ff:fea9:ee84	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
336	4.259025	fe80::6039:f6ff:fe7b:b087	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
337	4.259047	fe80::24c5:4ff:fe8a:faeb	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
338	4.259258	fe80::609:73ff:feaa:8b80	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
339	4.259259	fe80::74a8:deff:fe8b:4aa	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
340	4.259259	fe80::e0a2:5fff:fe18:2fe8	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
341	4.259259	fe80::8461:e8ff:fec4:28e5	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
342	4.259259	fe80::40bc:f2ff:fec8:62dd	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
343	4.259486	fe80::609:73ff:feaa:8ac0	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…
346	4.262062	fe80::10ae:ceff:fe68:2c1a	fe80::4e52:62ff:fe0e:5401	ICMPv6	94	Echo (ping) reply id=0x000…

Es lassen sich die folgenden Laborrechner identifizieren:
is-pc2, is-pc04, is-pc03, is-pc01, is-pc05, is-pc06 (jeweils durch die EUI-64 codierte MAC-Adresse)

Wie viele unterschiedliche Stationen antworten darauf, oder wie viele aktive Komponenten im RN-LAN arbeiten bereits mit IPv6?

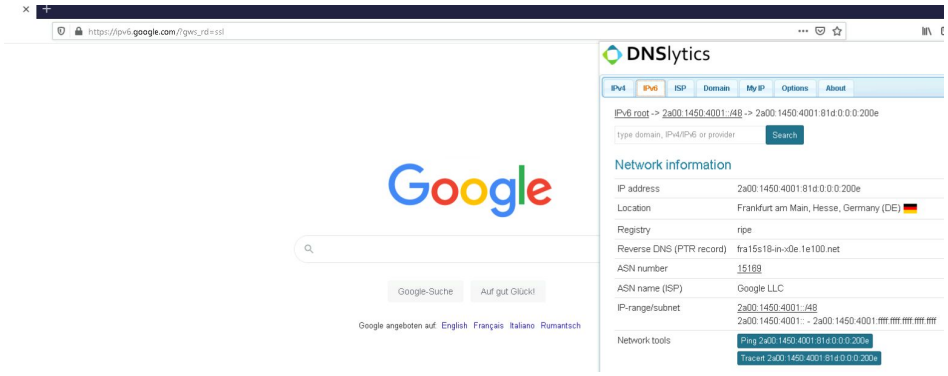
24 unterschiedliche Stationen antworten auf den Ping-Request.

Aufgabe 3

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?
Über ICMPv6 wird ein Router-Solicitation-Paket im Netzwerk versendet. Der Router antwortet hierauf mit einem Router-Advertisement. In Aufgabe 4 gehen wir ausführlich darauf ein.

Kommen Sie raus in das Internet? Was ist dazu noch erforderlich?
DNS-Server muss konfiguriert sein, um Namen auflösen zu können.

Rufen Sie die Webseite www.kame.net mittels IPv6-Adresse auf (kame.net ist manchmal instabil, alternativ versuchen Sie ipv6.google.com.



In Firefox ist das Add-On „IP Address and Domain Information“ installiert. Dieses Add-On zeigt ihnen oben rechts im Browser-Fenster die IP-Adresse (IPv4 und IPv6) der aufgerufenen Seite an.

- Mit welcher IPv6-Adresse sie im Netz unterwegs sind, zeigt die Seite http://www.heise.de/netze/tools/meine-ip-adresse an.
  142.62.66.1 → 2001:0470:0026:05f7:4e52:62ff:fe0e:e0e6
- **Welche IPv6-Adresse hat http://www.google.com ?**
  2a00:1450:4001:801::200e
- **Was ist das besondere an der IPv6-Adresse von Facebook?**
  2620:0:1cfe:face:b00c::3
  Facebook hat sich in seiner eigenen IPv6-Adresse als “Text” verewigt :)
- **Lösen Sie mittels nslookup oder dig die URL sixy.ch in die IPv6-Adresse auf!**

```
C:\Windows\System32>nslookup sixy.ch
Server:  opnsense.rnlabor.hdm-stuttgart.de
Address:  141.62.66.250

*** sixy.ch wurde von opnsense.rnlabor.hdm-stuttgart.de nicht gefunden: Non-existent domain.

C:\Windows\System32>nslookup sixy.ch
Server:  dns.google
Address:  2001:4860:4860::8888

*** sixy.ch wurde von dns.google nicht gefunden: Non-existent domain.
```

  Auch nach Eintragung des DNS-Servers von Google an der IPv6-Verbindung kann die Domain nicht aufgelöst werden.

Aufgabe 4

Neighbor Discovery (ND) ersetzt ARP. Dafür zuständig ist das RFC4861. Dabei handelt es sich um Neighbor Discovery for IP version 6 (IPv). Es wird in SLAAC benutzt. Dabei steht SLAAC für Stateless Address Autoconfiguration. Dabei handelt es sich um ein Verfahren um IPv6 zustandslos (stateless) automatisch zu konfigurieren. Dabei hat das ICMPv6 fünf Message Typen.

1. Router Advertisement (Ich bin Router A)
2. Router Solicitation (Host erzähl mir mehr über dich Router A)
3. Neighbor Advertisement (Ich bin Host 1 -> Host 2)
4. Neighbor Solicitation (Erzähl mir mehr über dich Host1, fragt Host 2)
5. Redirect (Ich kenne einen besseren Weg Router A <-> Router B)

Router Advertisement (RA)

- RA ist sehr wichtig für SLAAC
- wird zu bestimmten Intervallen versendet
- Unsolicited RA werden an FF02::1 gesendet
- Erhält Host Update Konfigurationen
- RA reagiert auch auf Router Solicitation (RS)

Dabei enthält es verschiedene Daten, Beispiele dazu in der Klammer.

- Stateless / stateful (DHCPv6)
- Netzwerk Präfix (2001:: /64)
- Standard Router (Standard Router für 200 Sekunden)
- Hop Limit (126)
- MTU (4096)

Router Solicitation (RS)

- Wird während des SLAAC versendet
- es wird eine unmittelbare Reaktion benötigt
- wird insgesamt 3x bei keinem response verschickt

Dabei enthält auch das RS verschiedene Daten, beispielsweise

- Quell Adresse
- Ziel Adresse
- ICMPv6 Type

Neighbor Advertisement (NA)

- Als Antwort auf ein Neighbor Solicitation
- oder falls das eigene NIC sich ändert
- enthält link-layer Adressen

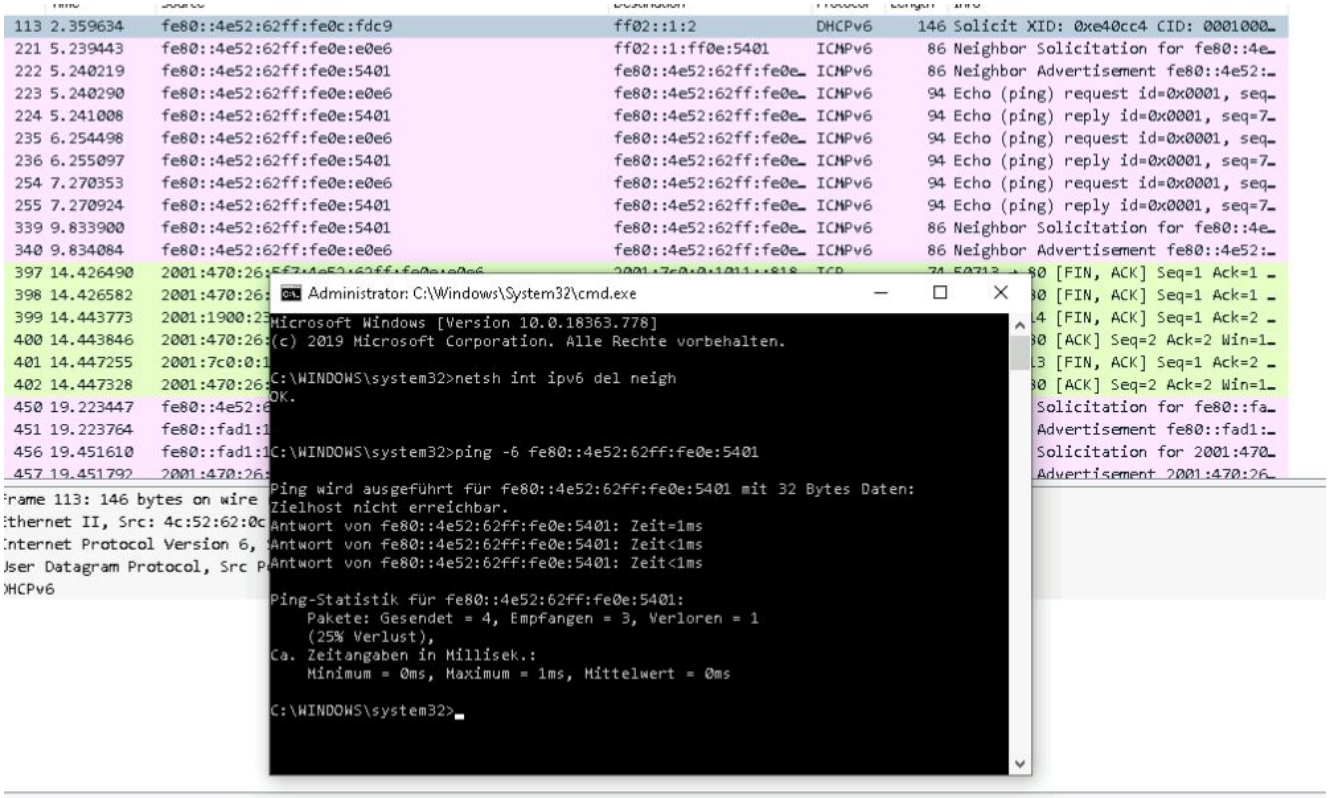
Daten

- ICMP Type 136 (https://tools.ietf.org/html/rfc4861)

Neighbor Solicitation (NS)

- NS fordert Informationen an
- Ist eine Neighbor Advertisement Antwort
- wird während dem SLAAC gesendet
- wird verwendet um die Erreichbarkeit zu verifizieren

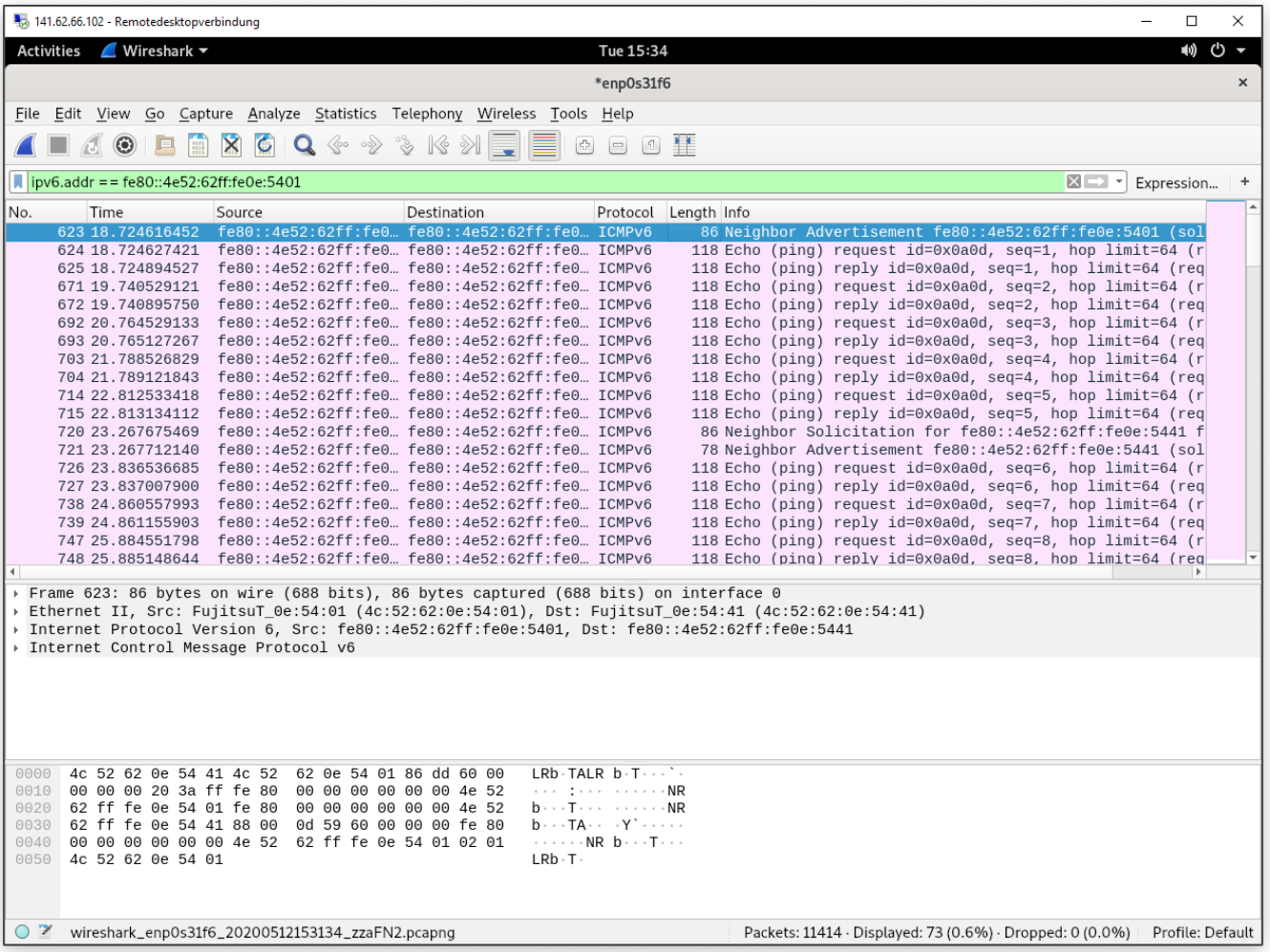
Ab hier kommt das Praktische Beispiel.
Nach dem Löschen des Neighbour Caches lässt sich folgendes erkennen:
Welches Protokoll/Type wird anstatt ARP zur Ermittlung der MAC-Adressen verwendet?
Anstatt von ARP wird ICMPv6 verwendet.
Windows:



Welche Zieladresse wird im ersten Neighbour-Paket verwendet und um welchen Adresstyp handelt es sich?
ff02::1:ff0e:5401 → Es ist eine Multicast-Adresse, die alle Nodes im Netzwerk abfragt.

Linux

Nachdem eingeben von ip neigh flush dev enp0s31f6 und dem schließenden ping6 fe80::4e52:62ff:fe0e:5401%enp0s31f6 passiert folgendes:



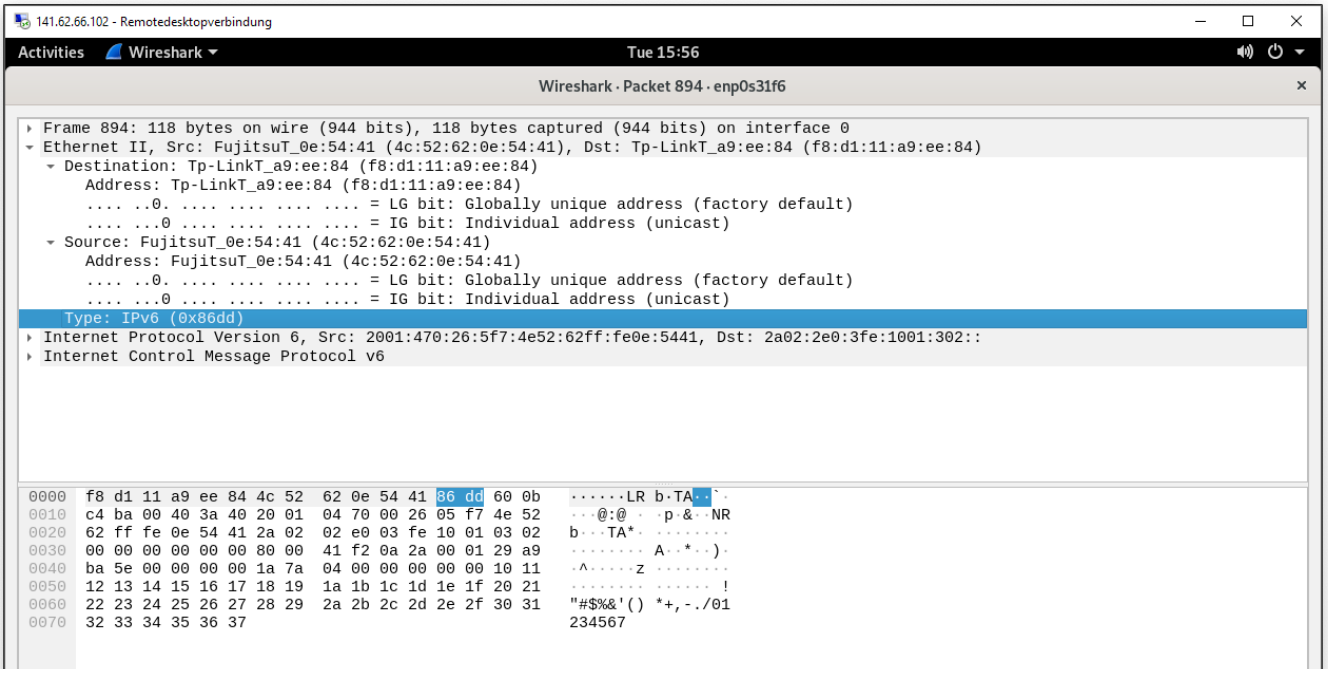
Man erkennt, dass über ICMPv6 (siehe Aufgabe 2 zu den Hintergründen) ein ICMPv6 NEIGHBOR ADVERTISEMENT Paket verschickt wird. Die Erklärung der einzelnen Pakete wurde am Anfang ausführlich beantwortet.

Aufgabe 5

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.

Wodurch wird im Ethernet-Frame auf das eingepackte IPv6 hingewiesen?

Unter Ethernet II in Wireshark wird der Ethernet Type auf IPv6 gesetzt.



Welche Bedeutung haben folgende Felder des IPv6-Headers und gibt es Entsprechungen in IPv4?

Version - 4 Bit - auch bei IPv4

Hier ist die Version des IP-Protokolls abgelegt, nach der das IP-Paket erstellt wurde.

Traffic Class - 8 Bit - auch bei IPv4 (Urgent Flag)

Der Wert des Feldes definiert die Priorität des Pakets.

Flow Label - 20 Bit

Kennzeichnet Pakete für ein viel schnelleres Routing.

Payload Length - 16 Bit

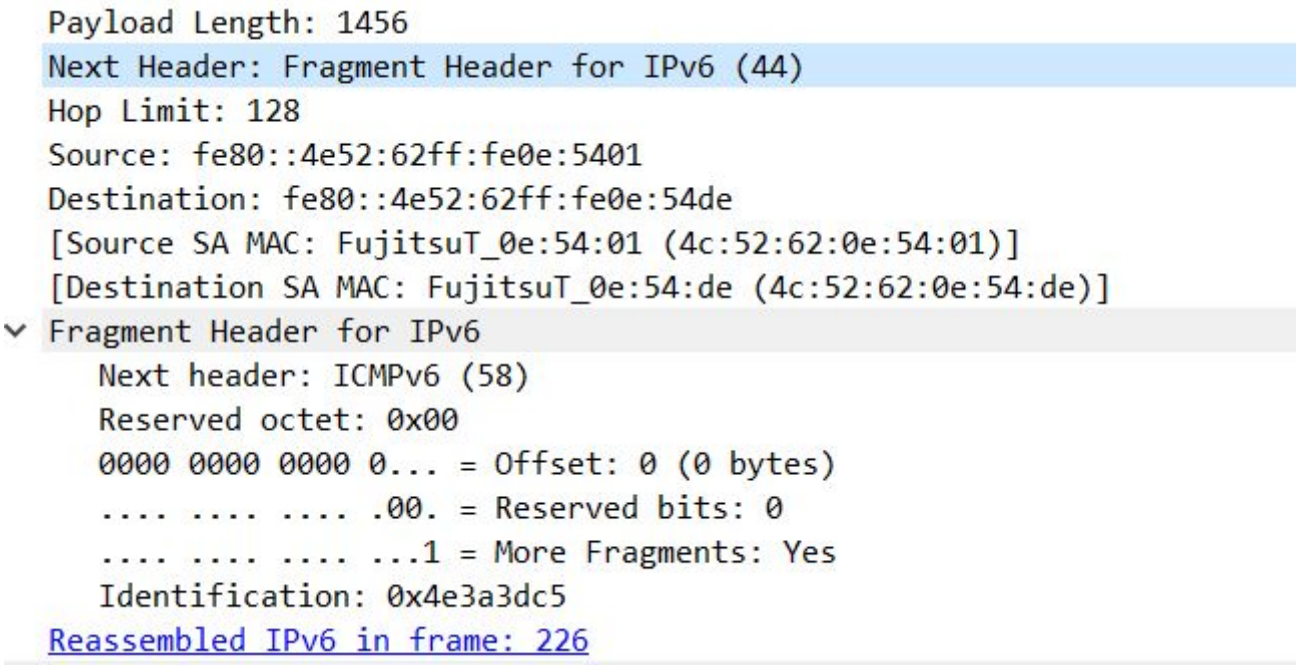
Hier steht die im IP-Paket transportierte Größe des Payloads.

Hop Limit - 8 Bit - auch bei IPv4

Enthält den TTL Wert, der nach jedem Hop dekrementiert.

Senden Sie nun ein 5000 Byte großes Paket vom Windows-PC an den Ubuntu-PC und schauen sich die Abfolge der Pakete an. Welcher Wert taucht im Next-Header-Feld Ihres IPv6 Headers auf?

Im Next-Header-Feld des IPv6-Headers wird der Wert “Fragment Header for IPv6 (44)” angezeigt. Im Header direkt wird nur die Nummer 44 mitgesendet, welche für den Fragment Header steht, WireShark fügt dies automatisch mit ein.
Bei IPv6 wurde das Feld für “Fragment Offset” nicht in den Header übernommen, welches Routern erlaubt, IP-Pakete zu fragmentieren. Somit kann bei IPv6 nur der Absender eine Fragmentierung vornehmen. Wird ein zu großes Paket an einen Router gesendet, so wie im Versuch das Paket mit 5000 Byte, so meldet dieser mithilfe des ICMPv6-Protokolls, dass das Paket zu groß ist (Packet too big). Nun kann der Sender entweder kleinere Pakete versenden oder das Paket mithilfe des Next-Headers “Fragment Header” fragmentieren.



Welche Bedeutung haben die unterschiedlichen Felder des Fragmentation Headers und wie setzt IPv6 die Pakete wieder zusammen?

Das ursprüngliche Paket (Original-Paket) besteht aus zwei Teilen:

- nicht fragmentierbarer Teil: IPv6 Header und Erweiterungs-Header, die von Knoten entlang der Route ausgewertet werden müssen
- fragmentierbarer Teil: Rest des Pakets, also alle Erweiterungs-Header, die erst vom Zielknoten verarbeitet werden müssen, und die Header und Daten der höheren Layer

Der fragmentierbare Teil des Pakets wird nun in Fragmente zerlegt. Jedes Fragment ist ein Vielfaches von 8 Bytes lang (außer möglicherweise dem letzten). Jedes Fragment wird nun als einzelnes Paket zusammen mit dem nicht-fragmentierbaren Teil und dem Fragment Header versendet. Der Fragment Header besteht aus den folgenden Feldern:

Feld	**Beschreibung**
Next Header	Header-Typ des fragmentierten Teils des Original-Pakets
Reserved	8-bit reserved Feld: Wird mit 0 für die Übertragung initialisiert und beim Empfänger ignoriert.
Fragment Offset	Offset (Verschiebung) der Daten, welche diesem Header folgen. Wird relativ zum Start des fragmentierten Teils des Original-Pakets in 8-Bytes-Einheiten angegeben. Markiert also die Stelle, an welche die Daten gehören, welche in diesem Fragment mitgesendet werden.
Reserved bits	2-bit reserved Feld: Wird mit 0 für die Übertragung initialisiert und beim Empfänger ignoriert.
More Fragments (M Flag)	1 - mehr Fragmente kommen noch 0 - letztes Fragment
Identification	Für jedes fragmentierte Paket wird vom Sender ein Identification Wert generiert. Dieser muss sich von allen kürzlich gesendeten fragmentierten Paketen unterscheiden, welche die gleiche Quell- und Zieladresse haben. Durch diese Nummer weiß der Empfänger, welche Fragmente zu welchem Original-Paket gehören (jedes Fragment eines Pakets hat die gleiche Identification)

Der Empfänger kann nun mithilfe des Fragment-Offsets und des M Flags die einzelnen Fragmente wieder zum Original-Paket zusammensetzen. Es werden nur Fragmente mit gleicher Quelladresse, Zieladresse und Identification Nummer zu einem Paket zusammengesetzt.

Aufgabe 6

IPv6 Adressen sind durch ihren Interface Identifier nachverfolgbar, deswegen wurden die Privacy Extensions eingeführt, die die Kopplung von Interface Identifier und MAC-Adresse aufheben und zufällig neue Interface Identifier erzeugen, die regelmäßig gewechselt werden. Dadurch wird dem Datenschutzproblem entgegengewirkt, sodass die genutzten Dienste wie z.B. Facebook die Benutzer nicht tracken können und dementsprechend personalisierte Werbung schalten.
Privacy Extensions ist eine Erweiterung für die Stateless Address Autoconfiguration (SLAAC) von IPv6, um den Hostanteil der IPv6-Adressen zu anonymisieren. Dies ist bei den meisten Betriebssystemen per Default normalerweise aktiviert, folgende Tabelle gibt Aufschluss darüber, ob man aktiv werden muss.

Betriebssystem	Privacy Extensions	ab Werk aktiv	de-/aktivierbar
Windows XP	+	+	+/+
Windows Vista	+	+	+/+
Windows 7	+	+	+/+
Windows Server 2003	+	-	+/+
Windows Server 2008	+	-	+/+
OpenSuse Linux	+	-	+/+
Ubuntu Linux	+	ab 12.04	+/+
Debian Linux	+	-	+/+
Fedora Linux	+	-	+/+
Mac OS X	+	ab 10.7	+/+
IOS 4.1	+	-	-/- (nur via Jailbreak)
IOS 4.2	+	-	-/- (nur via Jailbreak)
IOS 4.3	+	+	-/-
Android ab 2.1	+	-	-/- (nur via Rooting)

Da die Privacy Extensions in unserem Versuch deaktiviert sind, kann man sie wie folgt aktivieren.

Unter Windows:
netsh interface ipv6 set privacy state=enabled store=active
netsh interface ipv6 set privacy state=enabled store=persistent
netsh interface ipv6 set global randomizeidentifiers=enabled store=active
netsh interface ipv6 set global randomizeidentifiers=enabled store=persistent

Unter Linux:
Mittels eines Textverarbeitungsprogramms wie VI oder nano muss man mit root-Rechten die config-Datei "**/etc/sysctl.conf**" öffnen und die folgenden Werte abändern:

net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
net.ipv6.conf.enp0s31f6.use_tempaddr = 2

Die Ziffern am Ende geben letztlich den Status an:
0 = Privacy Extensions nicht aktiv
1 = temporäre Adresse bereitstellen
2 = temporäre Adresse bereitstellen und aktiv nutzen

Um die neuen Regeln letztlich im Kernel anzuwenden, müssen sie mittels **sysctl -p** aktiviert werden.

Während die Laborrechner erreichbar waren haben wir es leider versäumt einen geeigneten Screenshot zu machen und können deswegen nur die IPv6-Adresse nach Aktivierung der Privacy Extensions nennen.

```
2: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 100
0
    link/ether 4c:52:62:0e:54:de brd ff:ff:ff:ff:ff:ff
    inet 141.62.66.101/24 brd 141.62.66.255 scope global dynamic enp0s31f6
       valid_lft 8416sec preferred_lft 8416sec
    inet6 2001:470:26:5f7:11f3:a157:b76c:5b14/64 scope global temporary dynamic
       valid_lft 86246sec preferred_lft 14246sec
    inet6 2001:470:26:5f7:4e52:62ff:fe0e:54de/64 scope global dynamic mngtmpaddr
       valid_lft 86246sec preferred_lft 14246sec
    inet6 fe80::4e52:62ff:fe0e:54de/64 scope link
       valid_lft forever preferred_lft forever
```

Nun muss man noch die alte vergebene IP mit **ip addr del 2001:470:26:5f7:4e52:62ff:fe0e:54de** löschen.

Aufgabe 7

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

```
C:\WINDOWS\system32>netsh interface ipv6 set address "Ethernet 2" 2001:470:26:5f7:1234:12ff:fe34:1234
```

Warum sollten Sie jetzt alle übrigen IPv6-Adressen löschen?
Alle weiteren IPv6-Adressen sollten nun gelöscht werden, damit auch wirklich nur noch über die neu zugewiesene IP-Adresse kommuniziert werden kann. Ansonsten bleiben die anderen IP-Adressen ebenfalls aktiv:

```
C:\WINDOWS\system32>ipconfig

Windows-IP-Konfiguration


Ethernet-Adapter VirtualBox Host-Only Network #3:

   Verbindungsspezifisches DNS-Suffix:
   Verbindungslokale IPv6-Adresse  . : fe80::800:27ff:fe00:b%11
   IPv4-Adresse  . . . . . . . . . . : 192.168.56.1
   Subnetzmaske  . . . . . . . . . . : 255.255.255.0
   Standardgateway . . . . . . . . . :

Ethernet-Adapter Ethernet 2:

   Verbindungsspezifisches DNS-Suffix: rnlabor.hdm-stuttgart.de
   IPv6-Adresse. . . . . . . . . . . : 2001:470:26:5f7:1234:12ff:fe34:1234
   IPv6-Adresse. . . . . . . . . . . : 2001:470:26:5f7:4e52:62ff:fe0e:5401
   Verbindungslokale IPv6-Adresse  . : fe80::4e52:62ff:fe0e:5401%7
   IPv4-Adresse  . . . . . . . . . . : 141.62.66.9
   Subnetzmaske  . . . . . . . . . . : 255.255.255.0
   Standardgateway . . . . . . . . . : fe80::fad1:11ff:fea9:ee84%7
                                       141.62.66.250

C:\WINDOWS\system32>_
```

Angriffsvektoren vermeiden
Man sollte für den Dauerbetrieb eines Servers auf feste IPv6 Adressen, die auf dem System hinterlegt sind, besonderen Wert legen, beispielsweise im DNS und in Konfigurationsdateien, in Zugriffsbeschränkungen, Firewalls und Datenbanken. Dies sind mögliche Angriffsvektoren, die ausgeschlossen werden müssen. Daher sollten überschüssige IPv6 Adressen gelöscht werden.

Konfigurieren Sie die statische IPv6-Adresse über /etc/network/interfaces. Was wird dadurch verhindert?
Wenn man die IPv6 Adresse statisch in /etc/network/interfaces einträgt, hat es den Nachteil, dass man kein SLAAC (siehe Aufgabe 4 zu den Hintergründen) mehr benutzt. Kurz gesagt, verzichtet man auf den Vorteil, dass das hintere Ende durch die Privacy Extensions verschleiert wird. So ist man leichter über mehrere Netzwerke hinweg identifizierbar.

Aufgabe 8

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?

```
C:\WINDOWS\system32>netsh interface ipv6 show privacy
Der aktive Status wird abgefragt...

Parameter für temporäre Adressen
------------------------------------------------
Temporäre Adresse verwenden         : disabled
Versuch, doppelte Adr. zu entdecken : 3
Maximale Gültigkeitsdauer           : 7d
Maximale bevorzugte Gültigkeitsdauer: 1d
Regenerationszeit                   : 5s
Maximale Verzögerungszeit           : 10m
Verzögerungszeit                    : 6m36s
```

Halbieren Sie die "Maximale bevorzugte Gültigkeitsdauer" auf den Rechnern.

```
C:\WINDOWS\system32>netsh interface ipv6 set privacy maxpreferredlifetime=12h
OK.


C:\WINDOWS\system32>netsh interface ipv6 show privacy
Der aktive Status wird abgefragt...

Parameter für temporäre Adressen
------------------------------------------------
Temporäre Adresse verwenden         : disabled
Versuch, doppelte Adr. zu entdecken : 3
Maximale Gültigkeitsdauer           : 7d
Maximale bevorzugte Gültigkeitsdauer: 12h
Regenerationszeit                   : 5s
Maximale Verzögerungszeit           : 10m
Verzögerungszeit                    : 6m36s
```

Linux:
ipv6 set privacy maxpreferredlifetime=1d. - Maximale Gültigkeitsdauer

Stellen Sie den Zusammenhang zwischen Preferred Lifetime und Valid Lifetime anschaulich dar.
Wenn die Preferred Lifetime einer präferierten Adresse abläuft, so wird diese Adresse veraltet (deprecated). Danach kann und soll sie weiterhin als Quelladresse für bereits bestehende Kommunikationen dienen. Über die veraltete Adresse sollte jedoch keine neue Verbindung mehr aufgebaut werden, sofern eine andere, nicht veraltete Adresse zur Verfügung steht. Die veraltete Adresse ist also eine weiterhin valide Adresse des Interface. Wenn die Valid Lifetime ausläuft, wird die Adresse invalide. Danach darf sie nicht mehr für ein- und ausgehende Verbindungen genutzt werden.

Aufgabe 9

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

Da durch den Home-Office-Versuch IPv4 dauerhaft aktiviert sein muss, lässt sich der Versuch nur unter erschwerten Bedingungen durchführen. IPv4 kann nicht deaktiviert werden und so ist ein Windows-Update immer möglich. Nach etwas Recherche im Web sind wir auf folgenden Artikel gestoßen, der das Problem genauer untersucht.
https://insinuator.net/2014/05/microsoft-windows-update-over-ipv6-or-not/

Wie verhält sich Linux im Vergleich dazu? (Anmerkung: Mittels sudo apt-get update und sudo apt-get upgrade im Terminal lässt sich Linux updaten)

Solange man sich in einem IPv6 Netzwerk befindet, hat(te) der Paketmanager apt-get Probleme sich zu security.debian.org zu verbinden. Dies führt(e) zu einem Time-Out. Der einfachste Weg dies zu ändern ist, indem man apt-get sagt, welches Protokoll er nutzen soll.

Für IPv4:

apt-get -o Acquire::ForceIPv4=true update

Für IPv6:

apt-get -o Acquire::ForceIPv6=true update

Alternativ kann man es auch persistent in **/etc/apt/apt.conf.d/99force-ipv4** mit **Acquire::ForceIPv4 “true”;** speichern.