Lehrstuhl für Netzarchitekturen und Netzdienste
School of Computation, Information and Technology
Technische Universität München

Grundlagen Rechnernetze und Verteilte Systeme (IN0010)

Übungsblatt 8

16. Juni – 20. Juni 2026

Aufgabe 1 SLAAC & Das Neighbor Discovery Protocol

Abbildung 1.1 zeigt ein kleines Direktverbindungsnetz mit einem PC und einem Server und den jeweiligen MAC-Adressen. Allen Interfaces seien mittels SLAAC sowohl Link-Local (LL) als auch Global-Unique (GU) Adressen zugewiesen.

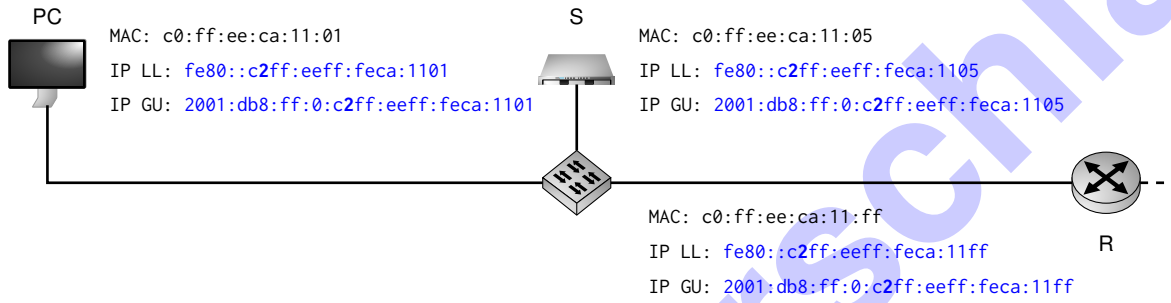


Abbildung 1.1: Netztopologie

Zunächst soll die Adressvergabe mittels SLAAC nachvollzogen werden.

a)* Bestimmen Sie die Link-Local-Adressen aller Interfaces.

- PC: `c0:ff:ee:ca:11:01` → `fe80::c2ff:eeff:feca:1101`
- S: `c0:ff:ee:ca:11:05` → `fe80::c2ff:eeff:feca:1105`
- R: `c0:ff:ee:ca:11:ff` → `fe80::c2ff:eeff:feca:11ff`

Hinweis: Das zweite Bit des ersten Oktetts jeder MAC-Adresse wird invertiert.
Grund: Manuell vergebene IPv6-Adressen haben häufig einen Interface-Identifier der Form `::abcd`, d. h. die ersten 48 bit sind 0. Schließt man nun von einer solchen IPv6-Adresse auf die zugrundeliegende MAC-Adresse, wäre das vorletzte Bit deren ersten Oktetts 0, was auf eine global eindeutige MAC-Adresse hinweisen würde – was offensichtlich falsch ist, da diese ja von einer manuell vergebenen IPv6-Adresse stammt.
Würde man dieses Bit nicht invertieren, müssten alle manuell vergebenen Interface-Identifier von der Form `2001:db8:1:0:200::1` sein (oder, falls die einmalige Abkürzung mehrerer Nullgruppen bereits im Subnet-Identifier lag, von der Form `2001:db8::200:0:0:1`).

b) Bestimmen Sie die Global-Unique-Adressen aller Interfaces. Nehmen Sie dazu an, dass Router R mit dem Präfix `2001:db8:ff::/64` konfiguriert ist und die Teilnehmer im Netz mittels *Router Advertisements* über das Präfix informiert.

Die Herleitung geschieht analog zu den Link-Local-Adressen, allerdings mit dem Präfix des jeweiligen Routers, welches über Router Advertisements bekannt gemacht wird.

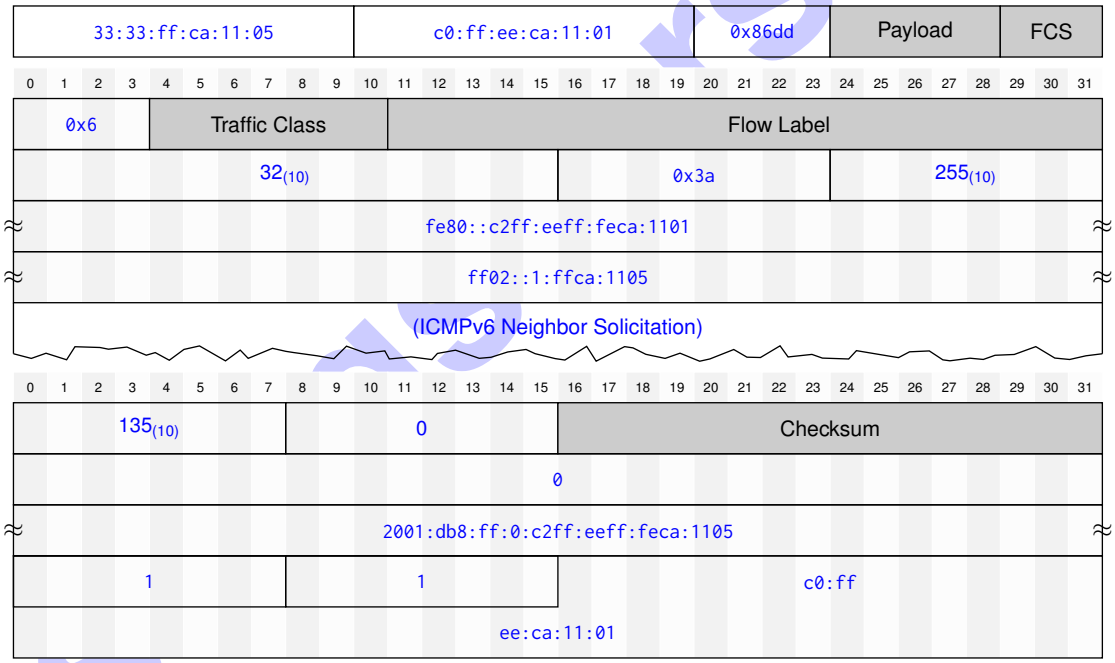
- PC: `c0:ff:ee:ca:11:01` → `2001:db8:ff:0:c2ff:eeff:feca:1101`
- S: `c0:ff:ee:ca:11:05` → `2001:db8:ff:0:c2ff:eeff:feca:1105`

Nun sendet der PC ein IPv6-Paket zum Server. Dabei wählt er die Global-Unique-Adresse des Servers als Zieladresse. Bevor aber ein Rahmen gesendet werden kann, muss, wie bei IPv4 auch, die Ziel-MAC-Adresse aufgelöst werden. Bei IPv4 wird dies durch ARP gelöst. Bei IPv6 hingegen kommt ein anderes Protokoll zum Einsatz: das Neighbor Discovery Protocol als Teil von ICMPv6.

c) Bestimmen Sie die Destination-MAC-Adresse der vom PC gesendeten *Neighbor Solicitation*.

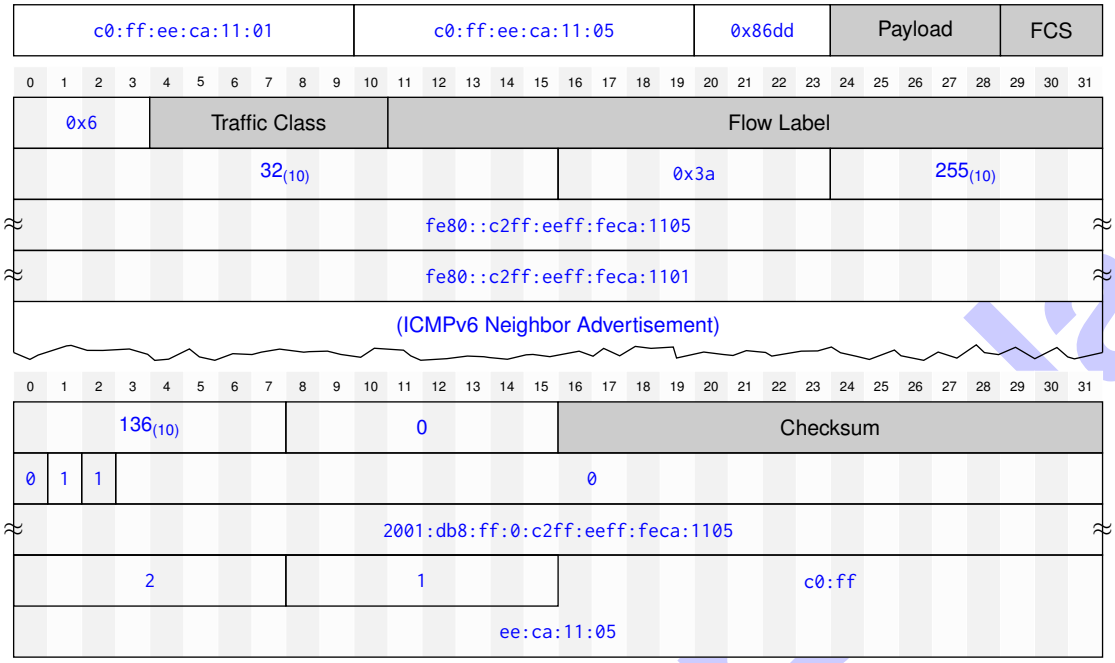
Neighbor-Solicitation-Pakete werden laut Vorlesung an die entsprechende Solicited-Node IPv6-Adresse geschickt. Da diese Multicast-Adressen sind, berechnet sich die Destination MAC-Adresse aus der Solicited-Node-Adresse.
Es soll die MAC-Adresse von S mit der IP `2001:db8:ff:0:c2ff:eeff:feca:1105` erfragt werden. Mit dem Solicited-Node-Prefix `ff02::1:ff00:0/104` und den letzten 24 Bit der Zieladresse ergibt sich die Solicited-Node-Adresse `ff02::1:ffca:1105`. Diese liegt im Prefix `ff00::/8` und ist somit eine Multicast-Adresse. Deshalb wird die zugehörige MAC aus dem Prefix `33:33` und den letzten 32 Bit der Mulitcast-Adresse `ff02::1:ffca:1105` berechnet und es ergibt sich `33:33:ff:ca:11:05`.

d) Füllen Sie für die Neighbor Solicitation die Ethernet-, IPv6- und Neighbor-Solicitation-Header aus.
Hinweis: Es ist nicht notwendig, die Header binär auszufüllen. Achten Sie aber darauf, dass Sie die Zahlenbasis deutlich kennzeichnen, z. B. `0x10` für hexadezimal oder $63_{(10)}$ für dezimal. Die Namen der Headerfelder und einige konstante Werte sind auf dem Cheatsheet abgedruckt.



Nach dem Erhalt der Neighbor Solicitation beantwortet der Server diese mit einem *Neighbor Advertisement*.

e) Füllen Sie nun für das Neighbor Advertisement die Ethernet-, IPv6- und Neighbor-Advertisement-Header aus.



Hinweise:

- Die Flags R (Router), S (Solicited) und O (Override) im Neighbor Advertisement werden wie folgt ausgefüllt:
  - R Nicht gesetzt, da das Advertisement von einem Host (Server S) und nicht von einem Router gesendet wird
  - S Gesetzt, da das Advertisement als Antwort auf eine Solicitation gesendet wird
  - O Gesetzt, da es sich nicht um eine Solicitation für eine Anycast-Adresse oder eine proxied Solicitation handelt
- Die *Target* Adressen in einem *Neighbor Advertisement* beziehen sich auf den Sender des Advertisements.
- Siehe auch RFC 4861 für weitere Informationen.

Aufgabe 2 Fragmentierung bei IPv6

In Abbildung 2.1 ist eine Anordnung von Netzkomponenten mit ihren MAC- und IP-Adressen dargestellt. Alle Adressen wurden mit SLAAC zugewiesen. Die Global-Unique-Adressen (GU) seien aus dem jeweiligen Präfix des Netzwerks abgeleitet. PC1 und PC2 nutzen die jeweiligen Router als Default Gateway.
PC1 sendet ein IPv6-Paket mit 1400 B Nutzdaten an PC2. Die MTU auf dem WAN-Link zwischen R1 und R2 betrage 1280 B[1]. Innerhalb der lokalen Netzwerke gelte die für Ethernet übliche MTU von 1500 B.

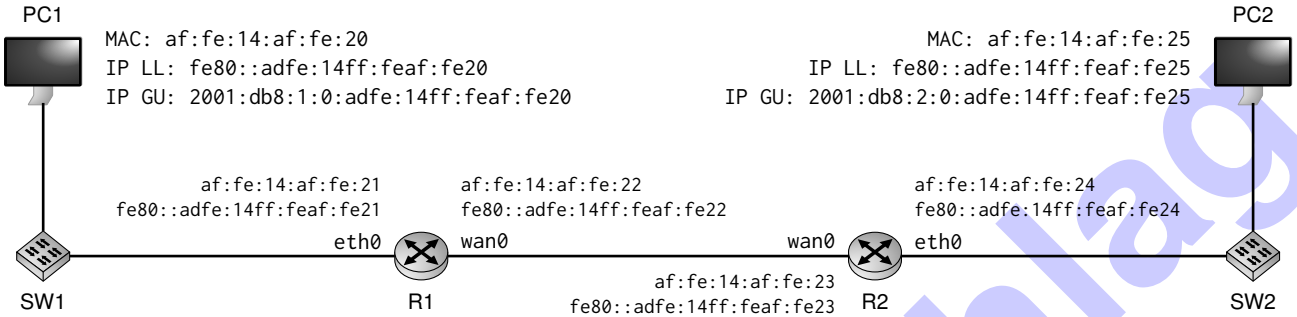


Abbildung 2.1: Netztopologie

a)* An welcher Stelle im Netzwerk wird die Fragmentierung stattfinden?

Direkt an PC1, da bei IPv6 keine Fragmentierung an Routern stattfindet.

b)* In wie viele Fragmente muss das Paket mindestens aufgeteilt werden?

Die MTU (Maximum Transmission Unit) ist die maximale Größe eines Pakets auf Schicht 3 inkl. Header. Sie entspricht also genau der maximalen Größe der Payload auf Schicht 2. Im Falle von Fragmentierung werden die einzelnen Fragmente jeweils einen IPv6-Header der Länge 40 B sowie einen Fragment-Header der Länge 8 B tragen. Sofern keine weiteren Extension-Header zum Einsatz kommen, erhalten wir demnach:

$$N = \left\lceil \frac{1400\,\mathrm{B}}{1280\,\mathrm{B} - 40\,\mathrm{B} - 8\,\mathrm{B}} \right\rceil = 2$$

c)* Begründen Sie, an welcher Stelle im Netzwerk die Fragmente reassembliert werden.

Erst der Empfänger, hier also PC2, reassembliert die Fragmente wieder. Tatsächlich kann i. A. kein anderer Knoten die Reassemblierung durchführen, da die Fragmente jeweils einzelne und voneinander unabhängige Pakete darstellen. Dies bedeutet insbesondere, dass sie unabhängig voneinander geroutet werden und daher u. U. verschiedene Wege zum Ziel nehmen können – das sieht man aus dem einfachen Beispiel in Abbildung 2.1 natürlich nicht, da es hier nur einen Pfad zwischen PC1 und PC2 gibt.

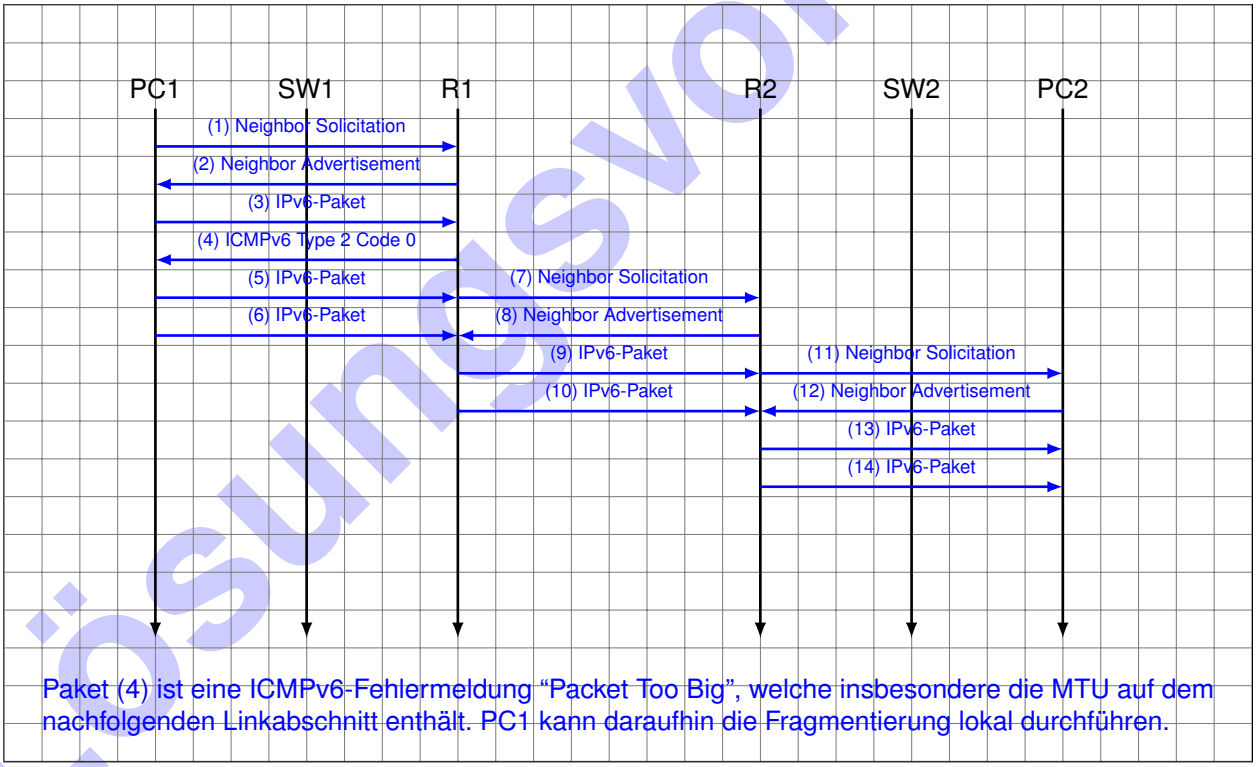
[1] Dies entspricht der minimalen MTU, die Schicht 2 laut RFC 2460 für IPv6 unterstützen muss.

d)* Vervollständigen Sie folgende Tabelle mit Informationen zu dem Paket bzw. den Fragmenten.

Fragment	Fragment Nutzdaten in Bytes	*Payload Length*	*Fragment Offset* in Bytes	*Fragment Offset* im Header	*More Fragments* Flag
Paket von PC1	1400	1400	nicht vorhanden	nicht vorhanden	nicht vorhanden
Fragment 1	1232	1240	0	0	1
Fragment 2	168	176	1232	154	0

Pro Fragment können 1280 B − 40 B − 8 B = 1232 B Nutzdaten übertragen werden. Da es sich dabei auch um ein Vielfaches von acht handelt (Fragment Offset ist in Vielfachen von 8 B angegeben), entspricht dies auch der tatsächlich übertragbaren Nutzdatenmenge.
Das erste Fragment hat daher 1232 B Nutzdaten und das zweite 168 B Nutzdaten. Für die im Header eingetragene Payload Length muss bei den Fragmenten die Länge der Extension-Headers (hier des Fragmentation-Extension-Headers) mit berücksichtigt werden.

e) Skizzieren Sie ein einfaches Weg-Zeit-Diagramm, welches **alle Rahmen** berücksichtigt, die auf den jeweiligen Verbindungen übertragen werden müssen. Nennen Sie die Art der ausgetauschten Rahmen und nummerieren Sie die Rahmen spaltenweise. Das Diagramm muss nicht maßstabsgetreu sein. Serialisierungszeiten und Ausbreitungsverzögerungen sind zu vernachlässigen.
Gehen Sie davon aus, dass derzeit keinerlei Mappings zwischen IP- und MAC-Adressen gecached sind.



Paket (4) ist eine ICMPv6-Fehlermeldung “Packet Too Big”, welche insbesondere die MTU auf dem nachfolgenden Linkabschnitt enthält. PC1 kann daraufhin die Fragmentierung lokal durchführen.

f) Geben Sie für jede *Neighbor Solicitation*, die Sie im Diagramm eingetragen haben, an, für welche IP-Adresse bzw. welches Interface die MAC-Adresse angefragt wurde.

- in (1): **R1.eth0**, da das Ziel nicht im lokalen Netz liegt und R1 das Default Gateway ist
- in (7): **R2.wan0**, da das Ziel nicht in einem direkt erreichbaren Netz liegt und R2 als Next-Hop konfiguriert ist.
- in (11): **PC2**, da das Ziel jetzt im lokalen Netz liegt, wird direkt die MAC-Adresse für die Ziel-IP-Adresse angefragt.

g) Füllen Sie für das erste **fragmentierte** IP-Paket, das von PC1 gesendet wird, den Ethernet- und IP-Header inkl. Extension-Header aus.
Hinweis: Es ist nicht notwendig, die Header binär auszufüllen. Achten Sie aber darauf, dass Sie die Zahlenbasis deutlich kennzeichnen, z. B. `0x10` für hexadezimal oder $63_{(10)}$ für dezimal. Die Namen der Headerfelder und einige konstante Werte sind auf dem Cheatsheet abgedruckt. Falls ein Wert nicht vorgegeben ist, treffen Sie eine sinnvolle Wahl.

5

af:fe:14:af:fe:21	af:fe:14:af:fe:20	0x86dd	Payload	FCS

5

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x6				Traffic Class							Flow Label																				
$1240_{(10)}$																$0x2c$								$64_{(10)}$						
≈	2001:db8:1:0:adfe:14ff:feaf:fe20																														≈
≈	2001:db8:2:0:adfe:14ff:feaf:fe25																														≈
(Fragment Header)																															

5

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x06								Reserved								0													Res.		1
0x12345678																															
1232 B Payload von Fragment #1																															

Hinweise:

- Der Wert für *Identification* kann frei gewählt werden, muss aber bei allen Fragmenten eines Pakets jeweils gleich sein und ändert sich über die Laufzeit des Fragments nicht.
- Als *Next Header* können z.B. TCP (`0x06`) und UDP (`0x11`) gewählt werden.

h) Begründen Sie, welche Headerfelder sich beim zweiten Fragment im Vergleich zum ersten Fragment ändern bzw. nicht ändern dürfen.

- **Identification**: muss bei allen Fragmenten eines Pakets gleich sein, damit diese als Teile desselben Pakets identifiziert werden können.
- **Payload Length**: das zweite Fragment ist mit nur noch 168 B Nutzdaten wesentlich kleiner.
- **Fragment Offset**: das zweite Fragment enthält einen anderen Teil der Nutzdaten.
- **More Fragments**: das zweite Fragment ist das letzte (zweite von zwei) Fragment. Es folgen keine weiteren Fragmente.

Andere Felder, insbesondere die Adressen, sind von der Fragmentierung nicht betroffen.

i) Begründen Sie, welche Headerfelder sich beim ersten Fragment im Rahmen der Weiterleitung ändern bzw. nicht ändern dürfen.

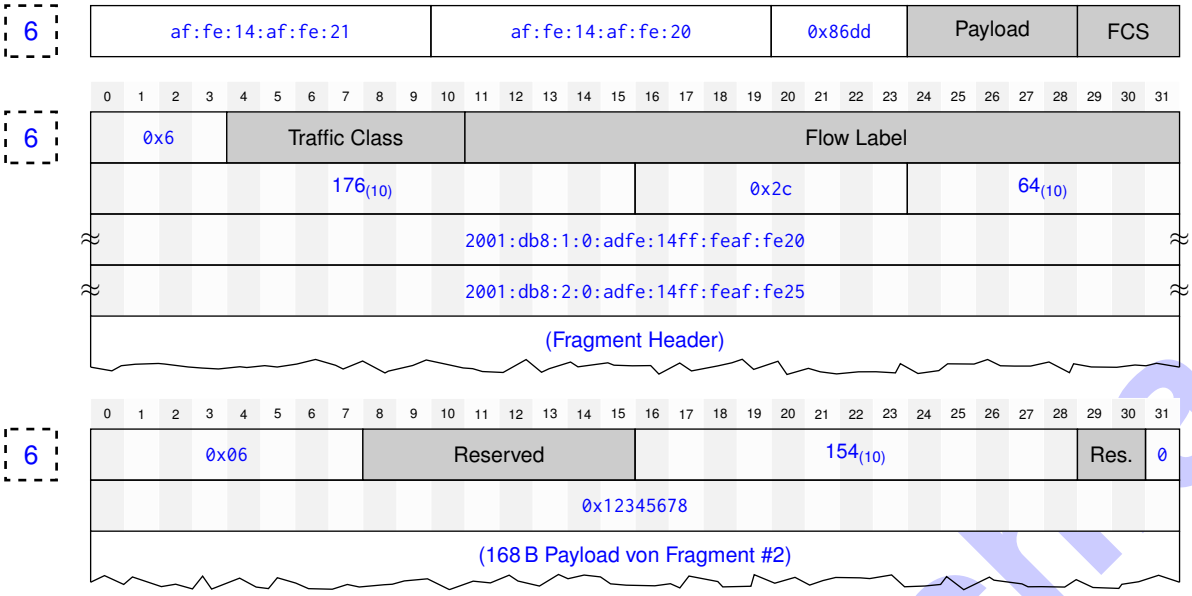
- **Adressen auf Schicht 2**: es werden immer die Interfaces im Direktverbindungsnetz adressiert (PC1-R1, R1-R2, R2-PC2).
- **Adressen auf Schicht 3**: Ende-zu-Ende-Adressierung. Die Adressen ändern sich nicht, sondern identifizieren vom Start bis zum Ziel Sender und Empfänger.
- **HopLimit**: wird bei jedem Durchlauf eines Routers um 1 verringert.

Andere Felder, insbesondere die Fragmentierungsinformationen, sind von der Weiterleitung nicht betroffen.

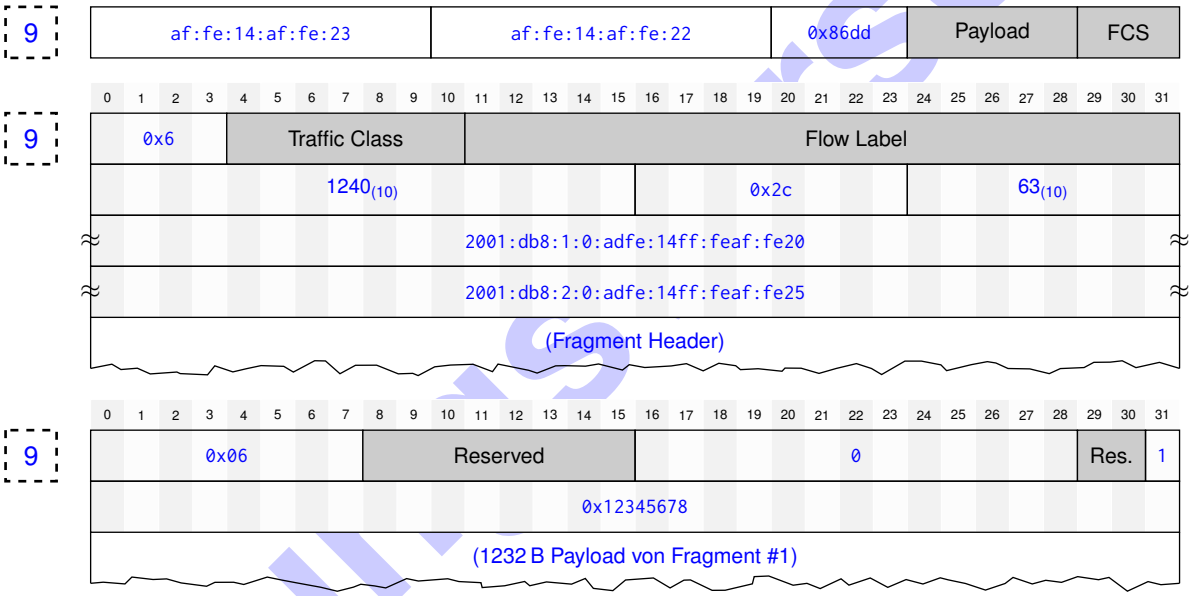
j) Fassen Sie die Unterschiede bei der Fragmentierung mit IPv4 und IPv6 zusammen.

Das grundlegende Prinzip der Fragmentierung ist bei IPv4 und IPv6 identisch.
Ein wesentlicher Unterschied ist zum einen der Ort der Fragmentierung: Bei IPv4 können Sender und Router fragmentieren; bei IPv6 kann dies nur beim Sender passieren. Ist ein Paket zu groß, wird ein ICMPv6-Fehler (Packet Too Big) erzeugt. Daher ist bei IPv6 die sogenannte Path MTU Discovery notwendig.
Außerdem wird die Fragmentierungsinformation nun nicht mehr im Header selbst, sondern in einem Extension-Header gespeichert. Dieser muss bei der Berechnung der maximal möglichen Nutzdaten pro Fragment entsprechend berücksichtigt werden.

k) **Übung:** Füllen Sie die Header für das **zweite Fragment** zwischen PC1 und R1 aus.



l) **Übung:** Füllen Sie die Header für das erste Fragment zwischen R1 und R2 aus.



m) **Übung:** Füllen Sie die Header für das erste Fragment zwischen R2 und PC2 aus.