Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

Grundlagen Rechnernetze und Verteilte Systeme (IN0010)

Übungsblatt 8

14. Juni – 18. Juni 2021

Aufgabe 1 Neighbor Discovery Protocol und IP-Fragmentierung bei IPv6

In Abbildung 1.1 ist eine Anordnung von Netzkomponenten mit ihren MAC-Adressen dargestellt. PC1 und PC2 seien mittels SLAAC sowohl Link-Local (LL) als auch Global-Unique (GU) Adressen zugewiesen. Für letztere werde das Präfix `2001:db8:1::/64` (PC1/R1) bzw. `2001:db8:2::/64` (PC2/R2) verwendet.
PC1 sendet ein IP-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.

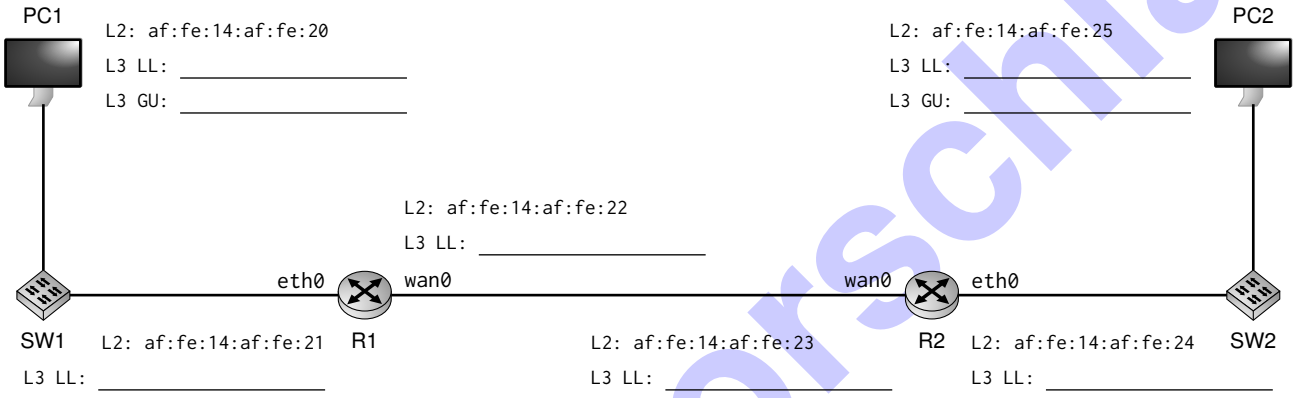


Abbildung 1.1: Netztopologie

Zunächst soll die Adressvergabe mittels SLAAC nachvollzogen werden.

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

Siehe Vorlesung:

- PC1: `af:fe:14:af:fe:20` → `fe80::adfe:14ff:feaf:fe20`
- R1.eth0: `af:fe:14:af:fe:21` → `fe80::adfe:14ff:feaf:fe21`
- R1.eth1: `af:fe:14:af:fe:22` → `fe80::adfe:14ff:feaf:fe22`
- R2.eth1: `af:fe:14:af:fe:23` → `fe80::adfe:14ff:feaf:fe23`
- R2.eth0: `af:fe:14:af:fe:24` → `fe80::adfe:14ff:feaf:fe24`
- PC2: `af:fe:14:af:fe:25` → `fe80::adfe:14ff:feaf:fe25`

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 hinweise 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-Identfier lag, von der Form `2001:db8::200:0:0:1`).

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

b) Bestimmen Sie die Global-Unique Adressen von PC1 und PC2. Nehmen Sie dazu an, dass Router R1 mit dem Präfix `2001:db8:1::/64` und Router R2 mit `2001:db8:2::/64` konfiguriert sind.

Die Herleitung geschieht analog zu den Link-Local-Adressen, allerdings mit dem Präfix des jeweiligen Routers, welche über Router Advertisements PC1 und PC2 bekannt gemacht werden.

- `af:fe:14:af:fe:20` → `2001:db8:1:0:adfe:14ff:feaf:fe20`
- `af:fe:14:af:fe:25` → `2001:db8:2:0:adfe:14ff:feaf:fe25`

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

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

d)* 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:

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

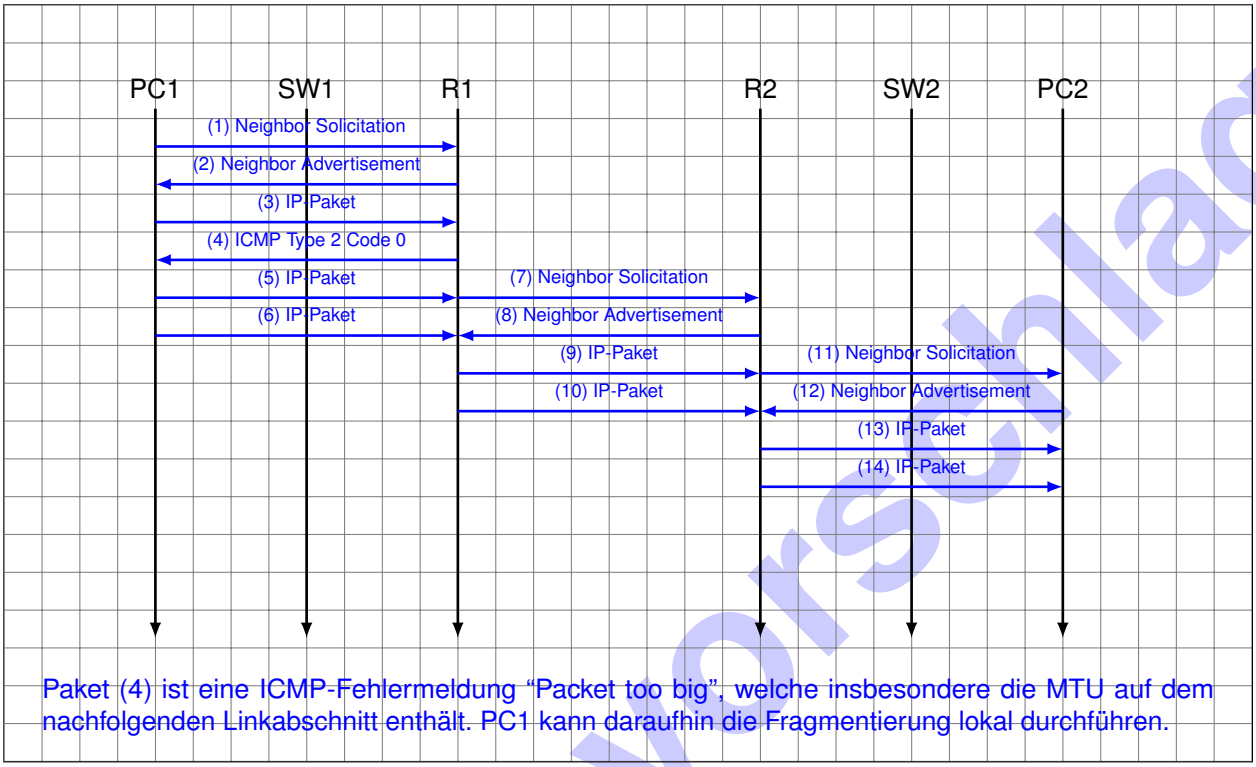
e) Bestimmen Sie die Größe der L3-SDU für jedes Fragment.

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 eine Payload von 1232 Bund das zweite eine Payload von 168 B.

f)* Begründen Sie, an welcher Stelle im Netzwerk werden 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 1.1 natürlich nicht, da es hier nur einen Pfad zwischen PC1 und PC2 gibt.

g) 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 geben Sie den Rahmen Nummern (1,2,3,...).** (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.
Nummerieren Sie die einzelnen Pakete Spaltenweise (Spalte $\triangleq$ Bereich z. B. zwischen R1 und R2).



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

h) Bestimmen Sie die Destination-MAC-Adresse des ersten übertragenen Rahmens.

Der Rahmen enthält ein IPv6 Neighbor Solicitation Paket. Diese werden laut Vorlesung an die entsprechende Solicited-Node Adress geschickt. Da diese Multicast Adressen sind, berechnet sich die Destination-MAC-Adresse aus der Solicited-Node Adresse.
Es soll die MAC Adresse für R1 mit der IP `fe80::adfe:14ff:feaf:fe21` erfragt werden. Mit dem Solicited-Node Prefix `ff02::1:ff00:0/104` und den letzen 24 bit der Ziel Adresse ergibt sich die Solicited-Node Adresse `ff02::1:ffaf:fe21`. Diese liegt im Prefix `ff00::/8` und ist somit eine Multicast Adresse. Deshalb wird die zugehörige MAC aus dem Prefix `33:33` und den letzen 32 bit der Ziel Adresse `ff02::1:ffaf:fe21` berechnet und es ergibt sich `33:33:ff:af:fe:21`.

Am Ende dieses Übungsblatts finden Sie Vordrucke für Ethernet-Header, ICMPv6 und IP-Header (mehr als benötigt). Es ist nicht notwendig, den Header binär auszufüllen. Achten Sie lediglich darauf, dass Sie die Zahlenbasis deutlich Kennzeichnen, z. B. `0x10` für hexadezimal oder $63_{(10)}$ für dezimal.

i) Füllen Sie für die ersten beiden Rahmen aus Teilaufgabe g) jeweils einen Ethernet- und einen IP-Header sowie die passende Payload aus. Beschriften Sie die gestrichelte Box neben dem jeweiligen Header/Paket mit der jeweiligen Rahmennummer.
Hinweis: Nutzen Sie den Cheatsheet zum bestimmen der Werte (z. B. Next Header). Sollte ein Wert nicht eindeutig bestimmt sein, treffen Sie eine sinnvolle Wahl.

j) Füllen Sie pro Pfadabschnitt (z. B. zwischen R1 und R2) für das jeweils erste fragmentierte Paket jeweils einen Ethernet- und einen IP-Header aus. Beschriften Sie die gestrichelte Box neben dem jeweiligen Header/Paket mit der jeweiligen Rahmennummer.
Hinweis: Nutzen Sie den Cheatsheet zum bestimmen der Werte (z. B. Next Header). Sollte ein Wert nicht eindeutig bestimmt sein, treffen Sie eine sinnvolle Wahl.

Aufgabe 2 Drahthai

Gegeben sei der in Abbildung 2.1 dargestellte Hexdump in Network-Byte-Order eines Ethernet-Rahmens, ohne Checksum, welcher im Folgenden analysiert werden soll.

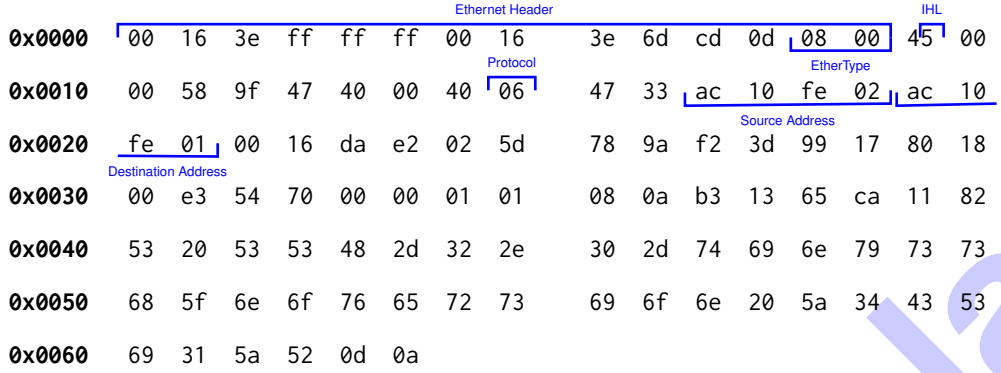


Abbildung 2.1: Hexdump eines Ethernet-Rahmens, ohne Checksum, in Network-Byte-Order

Hinweis: Zur Lösung der Aufgabe sind Informationen aus dem Cheatsheet notwendig.

a)* Markieren Sie in Abbildung 2.1 Beginn und Ende des Ethernet-Headers.

b) Begründen Sie, durch Markieren und Beschreiben relevanter Headerfelder, welches Protokoll auf Schicht 3 verwendet wird.

Der Ethertype gibt den Typ der Layer 2 Payload an. Der hier verwendete Wert `0x0800` steht für IPv4.

c)* Beschreiben Sie, wie die Länge des Headers auf Schicht 3 bestimmt wird. Markieren und benennen Sie dafür relevante Abschnitte in Abbildung 2.1.

Die Headerlänge in IPv4 wird durch das Headerfeld IHL angegeben. Dieses befindet sich im unteren Nibble des ersten Bytes des IPv4 Headers und gibt die Länge des Headers in Vielfachen von 4 B an. Die Länge des Headers beträgt also $5 \cdot 4\,\text{B} = 20\,\text{B}$.

d)* Markieren Sie alle Schicht 3 Addressen und benennen Sie diese.

e) Markieren Sie alle in Schicht 3 enthaltenen Extension Header.

Die Schicht 3 Payload ist IPv4. IPv4 kennt keine Extension Header sondern nur Optionen. Aus Teilaufgabe c) wissen wir, dass der Header 20 B lang ist, was auch der minimalen Länge des IPv4 Headers entspricht. Folglich ist nichts zu markieren.

f) Benennen und beschreiben Sie die drei kleinsten Headerfelder von Schicht 3. Geben Sie zudem die Größe der beschriebenen Headerfelder an.

Die drei kleinsten Headerfelder alle eine Größe von 1 bit.

RES `reserved`, reserviert um unter Umständen in Zukunft verwendet werden zu können

DF `do not fragment`, weißt den Verarbeitenden an, dass dieses Paket nicht fragmentiert werden darf

MF `more fragments`, informiert, dass — aufgrund einer vorangegangenen Fragmentierung — zu diesem IPv4 Paket weitere Fragmente gehören.

g) Falls es eine L3-SDU gibt, geben Sie ihren Typ an und begründen Sie die Angabe. Andernfalls, legen Sie Ihren Gedankengang dar und erörtern wie es zu dieser Situation kommen konnte.

Der Wert des IPv4 Headerfelds Protocol ist `0x06`. Demnach ist die L3-SDU TCP.

h) Die Bytes `0x0042` und Folgende sind Payload von Schicht 4. Geben Sie die ASCII Darstellung der ersten 7 B der Payload an.

Die ASCII Darstellung von `0x53 53 48 2d 32 2e 30` ist `SSH-2.0`.

i) Um welches Protokoll der Anwendungsschicht handelt es sich also vermutlich und wozu wird dieses Protokoll verwendet?

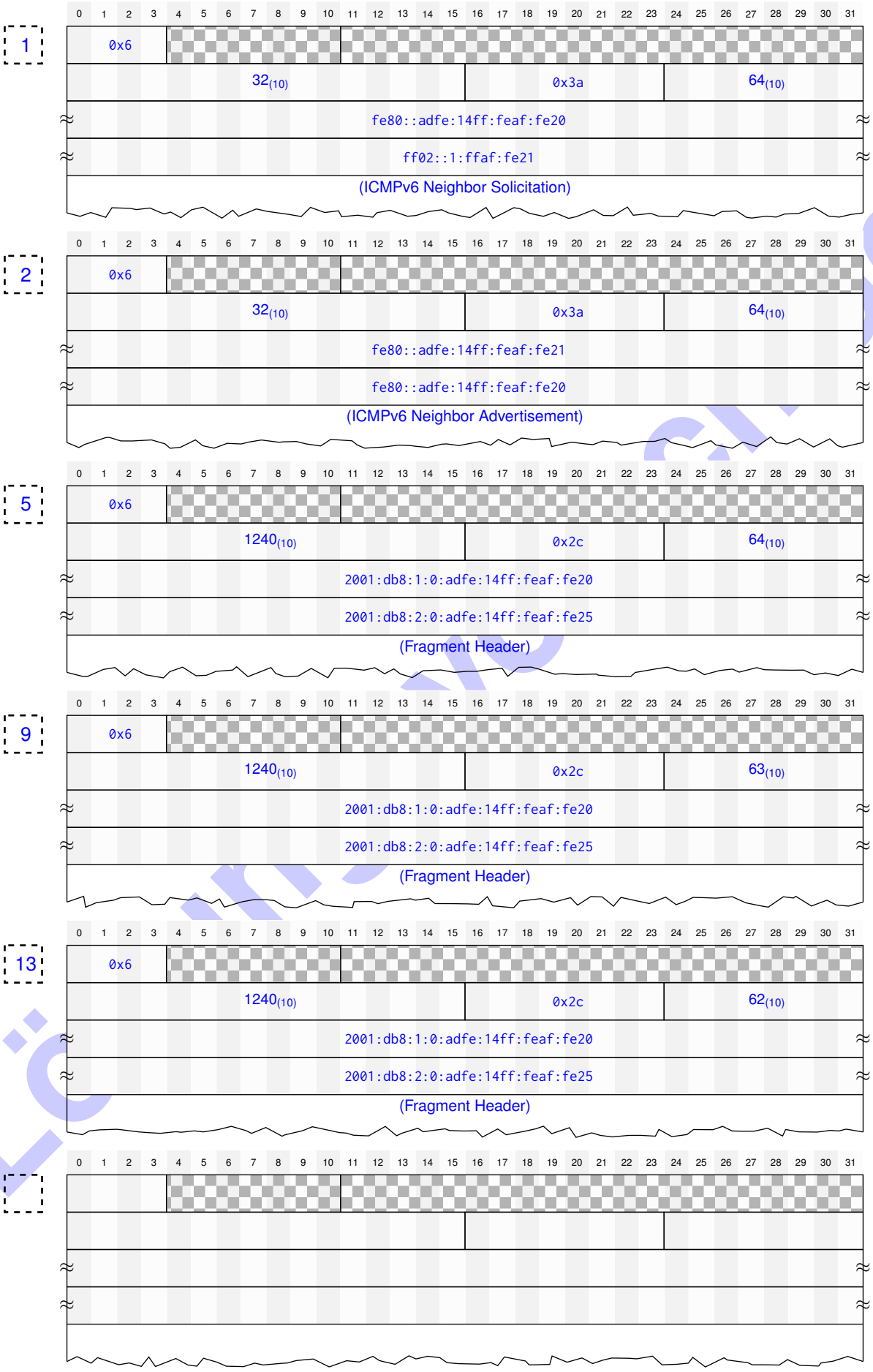
Es handelt sich um SSH (Version 2.0), das für eine verschlüsselte Konsolensitzung unter Linux/Unix und neuerdings auch unter Windows verwendet wird.

Vordrucke für Protokoll-Header:

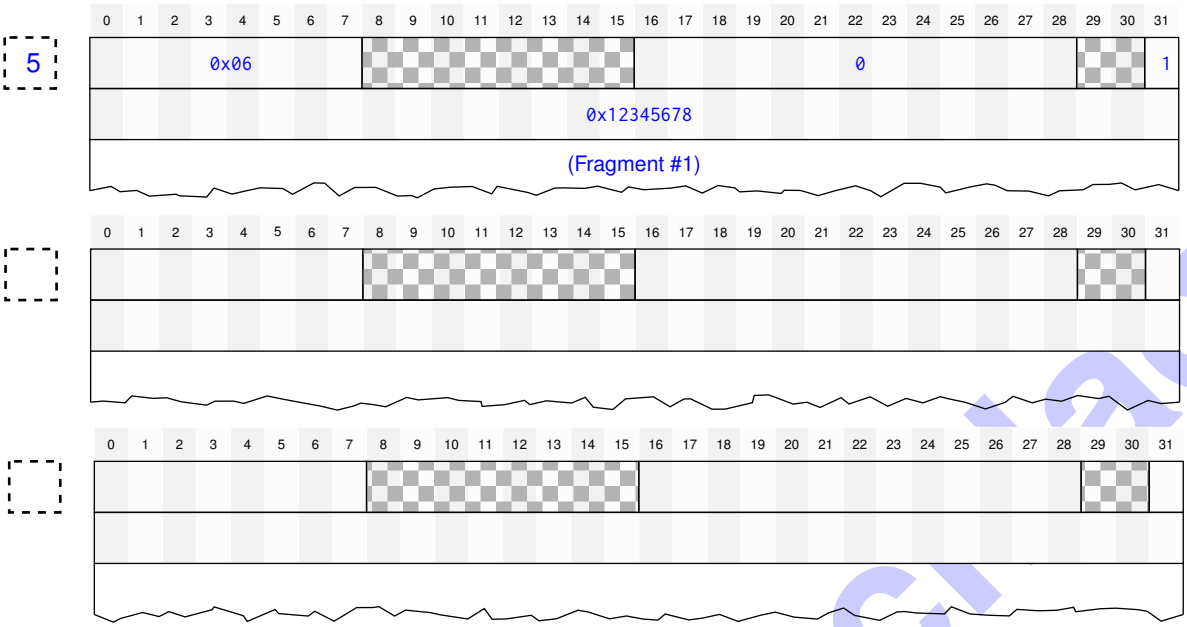
Ethernet-Frames

1	33:33:ff:af:fe:21	af:fe:14:af:fe:20	0x86dd	Payload	FCS
2	af:fe:14:af:fe:20	af:fe:14:af:fe:21	0x86dd	Payload	FCS
5	af:fe:14:af:fe:21	af:fe:14:af:fe:20	0x86dd	Payload	FCS
9	af:fe:14:af:fe:23	af:fe:14:af:fe:22	0x86dd	Payload	FCS
13	af:fe:14:af:fe:25	af:fe:14:af:fe:24	0x86dd	Payload	FCS
				Payload	FCS
				Payload	FCS
				Payload	FCS
				Payload	FCS
				Payload	FCS
				Payload	FCS
				Payload	FCS
				Payload	FCS
				Payload	FCS
				Payload	FCS

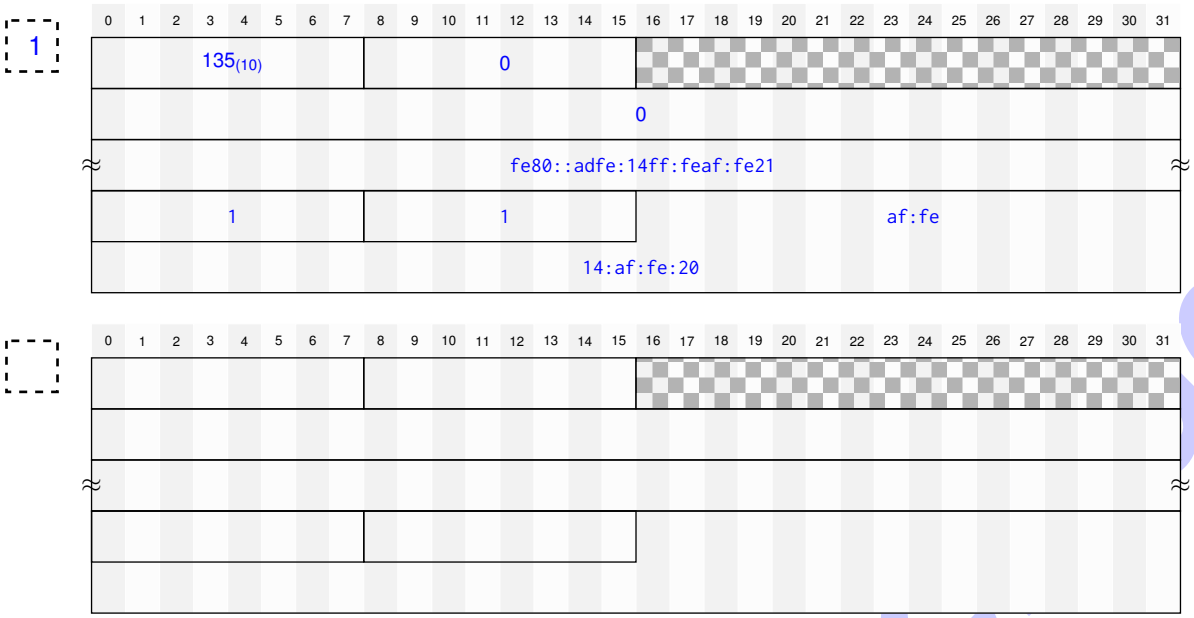
IPv6 Header


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
1
0x6
32(10)
0x3a
64(10)
fe80::adfe:14ff:feaf:fe20
ff02::1:ffaf:fe21
(ICMPv6 Neighbor Solicitation)
2
0x6
32(10)
0x3a
64(10)
fe80::adfe:14ff:feaf:fe21
fe80::adfe:14ff:feaf:fe20
(ICMPv6 Neighbor Advertisement)
5
0x6
1240(10)
0x2c
64(10)
2001:db8:1:0:adfe:14ff:feaf:fe20
2001:db8:2:0:adfe:14ff:feaf:fe25
(Fragment Header)
9
0x6
1240(10)
0x2c
63(10)
2001:db8:1:0:adfe:14ff:feaf:fe20
2001:db8:2:0:adfe:14ff:feaf:fe25
(Fragment Header)
13
0x6
1240(10)
0x2c
62(10)
2001:db8:1:0:adfe:14ff:feaf:fe20
2001:db8:2:0:adfe:14ff:feaf:fe25
(Fragment Header)

IPv6 Fragment Header



ICMPv6 Neighbor Solicitation



ICMPv6 Neighbor Advertisement