

Grundlagen Rechnernetze und Verteilte Systeme (IN0010)

Übungsblatt 7

9. Juni – 13. Juni 2026

Aufgabe 1 IP-Forwarding und Fragmentierung

In Abbildung 1.1 sind zwei PCs dargestellt, die in unterschiedlichen Netzen liegen. Diese Netze sind über die Router R1 und R2 miteinander verbunden. Die beiden Computer PC1 und PC2 verwenden jeweils den lokalen Router als Default-Gateway. Die Router seien entsprechend konfiguriert, sodass sie Pakete richtig weiterleiten¹. Neben allen Interfaces sind jeweils die MAC- und IP-Adressen des Interfaces angegeben. PC1 sendet ein IP-Paket mit 1000 B Nutzdaten an PC2. Die MTU auf dem WAN-Link zwischen R1 und R2 betrage 580 B. Innerhalb der lokalen Netzwerke gelte die für Ethernet übliche MTU von 1500 B.

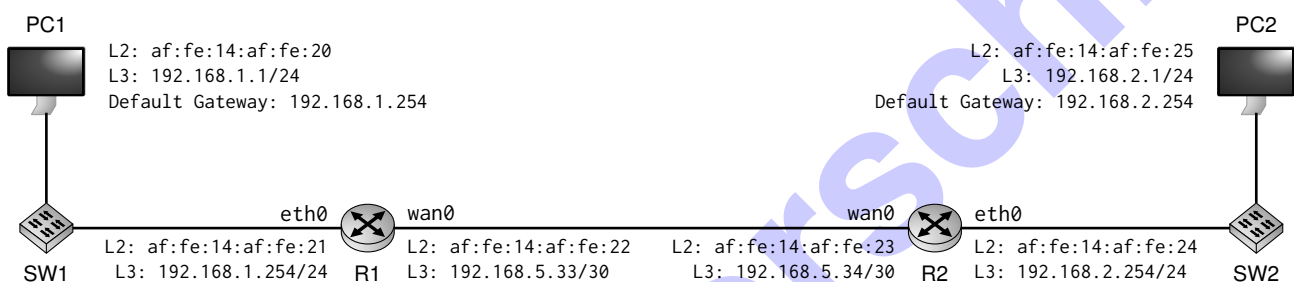


Abbildung 1.1: Netztopologie

Im Folgenden soll die Übertragung des Pakets mit allen notwendigen Zwischenschritten nachvollzogen werden. Gehen Sie zunächst davon aus, dass die ARP-Caches aller beteiligten Netzwerkkomponenten geleert sind.

a)* Inwiefern wirken sich die beiden Switches SW1 und SW2 in diesem Beispiel aus?

Die Switches haben keinerlei Einfluss auf die ausgetauschten Nachrichten.

Aufgrund der geringeren MTU zwischen den beiden Routern muss das von PC1 gesendete IP-Paket fragmentiert, also aufgeteilt werden.

b)* In wie viele Fragmente muss R1 das Paket von PC1 aufteilen?

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. Mit dem Wissen, dass ein IP-Header 20 B lang ist (Ausnahme bei Verwendung von Optionen), erhalten wir:

$$N = \left\lceil \frac{1000 \text{ B}}{580 \text{ B} - 20 \text{ B}} \right\rceil = 2$$

c)* An welcher Stelle im Netzwerk werden die Fragmente reassembliert?

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.

¹Wie genau ein Router Weiterleitungsentscheidungen trifft, sehen wir uns gegen Ende von Kapitel 3 an

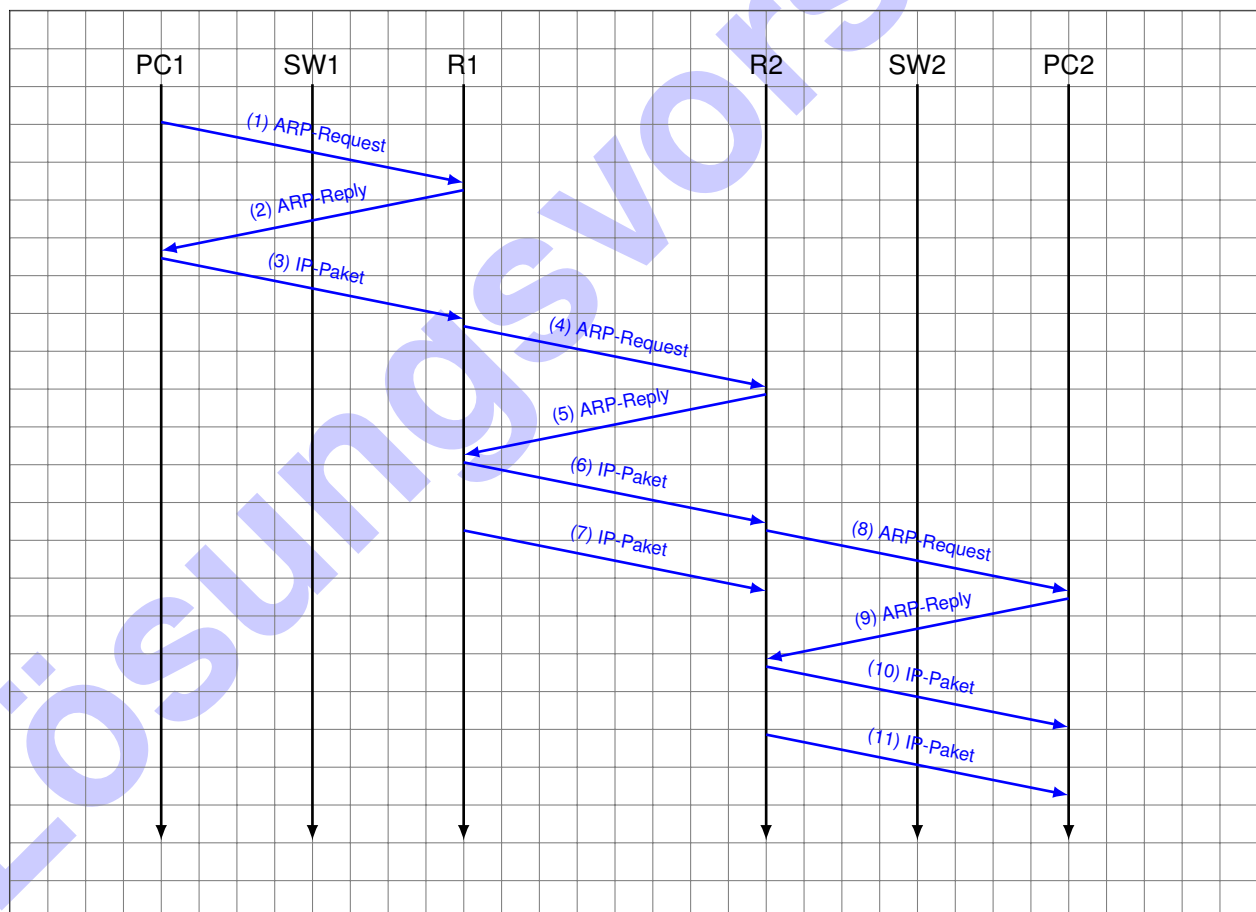
d)* Vervollständigen Sie folgende Tabelle mit Informationen zu dem Paket bzw. den Fragmenten.

Fragment	Fragment Nutzdaten in Bytes	Total Length	Fragment Offset in Bytes	Fragment Offset im Header	More Fragments Flag
Paket von PC1	1000	1020	0	0	0
Fragment 1	560	580	0	0	1
Fragment 2	440	460	560	70	0

Beachten Sie, dass in allen Fragmenten die Nutzdatenlänge ein Vielfaches von 8 sein muss, damit das *Fragment Offset* im nächsten Fragment korrekt als 8-fach Vielfaches angegeben werden kann. Das letzte Fragment ist davon ausgenommen, da es kein folgendes Fragment gibt. Dies ist hier implizit der Fall, muss aber bei anderen MTU Werten beachtet 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 geben Sie den Rahmen Nummern (1,2,3,...). Beachten Sie, dass die ARP-Caches aller Geräte leer sind.

Hinweis: Das Diagramm muss nicht maßstabsgetreu sein. Serialisierungszeiten und Ausbreitungsverzögerungen sind zu vernachlässigen.



f) Füllen Sie für das erste IP-Paket, das von PC1 gesendet wird den Ethernet- und IP-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.

3	af:fe:14:af:fe:21										af:fe:14:af:fe:20										0x0800				Payload				FCS			
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																																
3	0x4				0x5				ToS								1020 ₍₁₀₎															
32913 ₍₁₀₎																0 0 0				0 ₍₁₀₎												
64 ₍₁₀₎								Protocol								Hdr. Checksum																
192 ₍₁₀₎ 168 ₍₁₀₎ 1 ₍₁₀₎ 1 ₍₁₀₎																																
192 ₍₁₀₎ 168 ₍₁₀₎ 2 ₍₁₀₎ 1 ₍₁₀₎																																
Payload 1000B																																

g) Füllen Sie für alle IP-Pakete bzw. Fragmente zwischen R1 und R2 einen Ethernet- und IP-Header aus.

6	af:fe:14:af:fe:23										af:fe:14:af:fe:22										0x0800				Payload				FCS			
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																																
6	0x4				0x5				ToS								580 ₍₁₀₎															
32913 ₍₁₀₎																0 0 1				0 ₍₁₀₎												
63 ₍₁₀₎								Protocol								Hdr. Checksum																
192 ₍₁₀₎ 168 ₍₁₀₎ 1 ₍₁₀₎ 1 ₍₁₀₎																																
192 ₍₁₀₎ 168 ₍₁₀₎ 2 ₍₁₀₎ 1 ₍₁₀₎																																
Payload 560B																																

7	af:fe:14:af:fe:23										af:fe:14:af:fe:22										0x0800				Payload				FCS			
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																																
7	0x4				0x5				ToS								460 ₍₁₀₎															
32913 ₍₁₀₎																0 0 0				70 ₍₁₀₎												
63 ₍₁₀₎								Protocol								Hdr. Checksum																
192 ₍₁₀₎ 168 ₍₁₀₎ 1 ₍₁₀₎ 1 ₍₁₀₎																																
192 ₍₁₀₎ 168 ₍₁₀₎ 2 ₍₁₀₎ 1 ₍₁₀₎																																
Payload 440B																																

h) Füllen Sie für alle IP-Pakete bzw. Fragmente zwischen R2 und PC2 einen Ethernet- und IP-Header aus.

10	af:fe:14:af:fe:25	af:fe:14:af:fe:24	0x0800	Payload	FCS
----	-------------------	-------------------	--------	---------	-----

10	0x4 0x5 ToS	580 ₍₁₀₎
	32913 ₍₁₀₎	0 0 1 0 ₍₁₀₎
	62 ₍₁₀₎	Protocol Hdr. Checksum
	192 ₍₁₀₎ 168 ₍₁₀₎ 1 ₍₁₀₎ 1 ₍₁₀₎	
	192 ₍₁₀₎ 168 ₍₁₀₎ 2 ₍₁₀₎ 1 ₍₁₀₎	
	Payload 560B	

11	af:fe:14:af:fe:25	af:fe:14:af:fe:24	0x0800	Payload	FCS
----	-------------------	-------------------	--------	---------	-----

11	0x4 0x5 ToS	460 ₍₁₀₎
	32913 ₍₁₀₎	0 0 0 70 ₍₁₀₎
	62 ₍₁₀₎	Protocol Hdr. Checksum
	192 ₍₁₀₎ 168 ₍₁₀₎ 1 ₍₁₀₎ 1 ₍₁₀₎	
	192 ₍₁₀₎ 168 ₍₁₀₎ 2 ₍₁₀₎ 1 ₍₁₀₎	
	Payload 440B	

i)* Angenommen PC1 und PC2 würden über IPv6 kommunizieren:

1. Welche Auswirkungen hätte dies auf die Switches SW1 und SW2?
2. Müssten die Router R1 und R2 in diesem Fall auch IPv6-fähig sein?
3. An welcher Stelle fände die Fragmentierung von Paketen statt?

1. Im gegebenen Fall gar keine: Switches arbeiten nur mit MAC-Adressen, an denen sich nichts ändern würde (abgesehen von ggf. Multicast).
2. Ja, zumindest an den lokalen Interfaces eth0, da IPv6 und IPv4 nicht kompatibel sind. Ein Transport von IPv6 über IPv4 mittels GRE (General Routing Encapsulation) ist dann zwar theoretisch möglich, wegen der nicht-injektiven Abbildbarkeit von IPv6 auf IPv4 aber wenig sinnvoll bzw. im Allgemeinen unmöglich.
3. Fragmentiert würde nun direkt an PC1, da bei IPv6 Router grundsätzlich nicht fragmentieren.

Aufgabe 2 Subnetting

Der TUMexam AG werden die Adressbereiche 131.159.32.0/22 und 131.159.36.0/24 zugewiesen. Für die Aufteilung dieses Adressbereichs ist die TUMexam AG selbst verantwortlich. Nach einer sorgfältigen Bedarfsanalyse ergeben sich die folgenden Anforderungen an die Subnetze und die Mindestanzahl **nutzbarer** IP-Adressen:

Subnetz	NET 1	NET 2	NET 3	NET 4	NET 5
IPs	300	300	15	40	4

Bei der Erhebung dieser Zahlen wurde die an das jeweilige Router-Interface zu vergebende IP-Adresse bereits berücksichtigt.

a) Geben Sie jeweils die erste und letzte IP-Adresse der beiden vergebenen Adressbereiche an.

- 131.159.32.0/22:
Erste IP: 131.159.32.0 (Netzadresse)
Letzte IP: 131.159.35.255 (Broadcast-Adresse)
- 131.159.36.0/24:
Erste IP: 131.159.36.0 (Netzadresse)
Letzte IP: 131.159.36.255 (Broadcast-Adresse)

b) Wie viele IP-Adressen stehen der TUMexam AG insgesamt zur Verfügung? Können alle davon zur Adressierung von Hosts verwendet werden?

- 131.159.32.0/22: $2^{32-22} = 2^{10} = 1024$ Adressen
- 131.159.36.0/24: $2^{32-24} = 2^8 = 256$ Adressen

Insgesamt stehen also $1024 + 256 = 1280$ Adressen zur Verfügung. Allerdings sind die erste (Netzadresse) und letzte Adresse (Broadcast-Adresse) eines jeden Netzes nicht zur Adressierung von Hosts nutzbar. Es stehen also zunächst maximal $1022 + 254 = 1276$ Adressen zur Hostadressierung zur Verfügung.

c)* Ist es möglich, den von den beiden Adressblöcken gebildeten Adressbereich in einem einzigen Subnetz zusammenzufassen?

Nein. Die Subnetze sind nicht gleich groß (/22 und /24) und können damit keinesfalls zusammengefasst werden, da das nächst größere Subnetz mit einem /21 Präfix in jedem Fall weitere Netze enthalten würde.
(Ein einzelnes Subnetz hat als Größe immer eine Zweierpotenz, wir bräuchten hier aber eines mit $1024 + 256 = 1280$ Adressen.)

Achtung: Das obige Kriterium ist nur notwendig, nicht hinreichend! Zwei gleich große Subnetze können auch nur dann zusammengefasst werden, wenn sie aufeinander folgen **und** sich im nächst größeren Subnetz zusammen fassen lassen. (Letzteres Kriterium ist gleichbedeutend mit einem gemeinsamen Vaterknoten der beiden Subnetze, wenn man sich den Adressraum als Binärbaum vorstellt.)

d) Teilen Sie nun die beiden Adressbereiche gemäß der Bedarfsanalyse auf, so dass Subnetze der passenden Größe entstehen. Gehen Sie mit den Adressen so sparsam wie möglich um. Es soll am Ende ein möglichst großer zusammenhängender Adressbereich für zukünftige Nutzung frei bleiben. Tragen Sie die entsprechenden Werte und Adressen in die Tabelle ein.

Subnetz	NET 1	NET 2	NET 3
Bedarf	300	300	15
Größe	512	512	32
Nutzbar	510	510	30
Präfixnotation	131.159.32.0/23	131.159.34.0/23	131.159.36.64/27
Subnetzmaske	255.255.254.0	255.255.254.0	255.255.255.224
Netzadresse	131.159.32.0	131.159.34.0	131.159.36.64
Broadcast	131.159.33.255	131.159.35.255	131.159.36.95

Subnetz	NET 4	NET 5
Bedarf	40	4
Größe	64	8
Nutzbar	62	6
Präfixnotation	131.159.36.0/26	131.159.36.96/29
Subnetzmaske	255.255.255.192	255.255.255.248
Netzadresse	131.159.36.0	131.159.36.96
Broadcast	131.159.36.63	131.159.36.103

Um die Vorgaben zu erfüllen, müssen wir die Subnetze gemäß ihrer Größe in absteigender Reihenfolge bearbeiten. Andernfalls könnten wir die folgende Situation erhalten:

- An Netz 3 wird der Adressbereich 131.159.36.0/27 vergeben.
- Vergibt man nun aber an Netz 4 den Bereich 131.159.36.32/26, macht man einen Fehler. Um dies zu verstehen, muss man sich die Binärschreibweise der Netzadresse und Subnetz-Maske ansehen:
131.159. 36.0010 0000 (IP)
255.255.255.1100 0000 (Subnetz-Maske)
Eine UND-Verknüpfung beider Zeilen ergibt, dass die IP-Adresse 131.159.36.32 in das Subnetz 131.159.36.0/26 fällt!
- Wir müssten also den Bereich 131.159.36.64/26 an Netz 4 vergeben. Dann allerdings entstünde eine Lücke zwischen Netz 3 und Netz 4.
- Vergibt man die Adressen gemäß der Größe der Subnetze in absteigender Reihenfolge, umgeht man das Problem. Dieses Vorgehen könnte natürlich wieder anderen Kriterien widersprechen – beispielsweise der Vergabe zusammenhängender Adressblöcke an einzelne Niederlassungen.

Aufgabe 3 IPv6 & Supernetting (Zusatzaufgabe)

Der TUMexam AG wurden nun die IPv6 Adressebereiche $2001:0db8:0001:000d:0000:0000:0000:0000/64$ (*NET1*) und $2001:0db8:0001:000e:0000:0000:0000:0000/64$ (*NET2*) zugeteilt.

a)* Geben Sie die in *NET1* enthaltene IPv6 Adresse $2001:0db8:0001:000d:0000:00f0:0000:0000$ in kompakter Schreibweise an.

- führende Nullen werden weg gelassen: $2001:db8:1:d:0:f0:0:0$
- der größte konsekutive Block von mindesten 2 „Nuller“-Blöcken kann durch $::$ abgekürzt werden: $2001:db8:1:d:0:f0::$

b)* Wieviele Adressen enthält jedes Präfix?

$$2^{128-64} = 18\,446\,744\,073\,709\,551\,616 \approx 18,45 \text{ Trillionen}$$

c) Wie oft kann der gesamte IPv4 Adressbereich ($0.0.0.0/0$) in *NET1* abgebildet werden?

$$2^{(128-64)-32} = 2^{32} = 4\,294\,967\,296 \approx 4,29 \text{ Milliarden.}$$

d)* Welche Bedingungen müssen erfüllt sein, damit 2 Subnetze aggregiert werden können?

- gleich groß, d. h. selbe Präfixlänge n
- benachbart (auf die letzte Adresse im ersten Netz muss direkt das nächste Netz folgen)
- Es muss eine valide Präfixmaske mit Länge $n - 1$ existieren, d. h. die beiden Netze dürfen sich nur genau im letzten Bit ihres Präfix unterscheiden.

e)* Können die beiden Subnetze *NET1* und *NET2* in ein $/63$ Subnetz aggregiert werden?

Obwohl die Netze von gleicher Größe sind und nebeneinander liegen, können sie nicht aggregiert werden, da sie nicht im gleichen $/63$ Präfix liegen. Für die Bits 61 bis 64: $d_{16} = 1101_2$, $e_{16} = 1110_2$. $2001:db8:1:c::/62$ würde die beiden Netze umfassen, aber zusätzlich auch

- $2001:db8:1:c::/64$ und
- $2001:db8:1:f::/64$ enthalten.

