

## Tutorübung zur Vorlesung Grundlagen Rechnernetze und Verteilte Systeme

### Übungsblatt 5 (18. Mai – 22. Mai 2015)

**Hinweis:** Die mit \* gekennzeichneten Teilaufgaben sind ohne Kenntnis der Ergebnisse vorhergehender Teilaufgaben lösbar.

#### Aufgabe 1 Subnetting

Der Moepinet AG werden die Adressbereiche 131.159.32.0/22 und 131.159.36.0/24 zugewiesen. Für die Aufteilung dieses Adressbereichs ist die Moepinet 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.
- b) Wie viele IP-Adressen stehen der Moepinet AG insgesamt zur Verfügung? Können alle davon zur Adressierung von Hosts verwendet werden?
- c)\* Ist es möglich, den von den beiden Adressblöcken gebildeten Adressbereich in einem einzigen Subnetz zusammenzufassen?
- 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. Für jedes Subnetz ist anzugeben:
  - die Größe des Subnetzes
  - die Anzahl nutzbarer Adressen
  - das Subnetz in Präfixschreibweise
  - die Subnetzmaske in Dotted-Decimal-Notation
  - die Netz- und Broadcastadresse

#### Aufgabe 2 Shortest Path Trees (Hausaufgabe)

In dieser Aufgabe betrachten wir die aus der Vorlesung bekannten Algorithmen zur Konstruktion von SPTs.

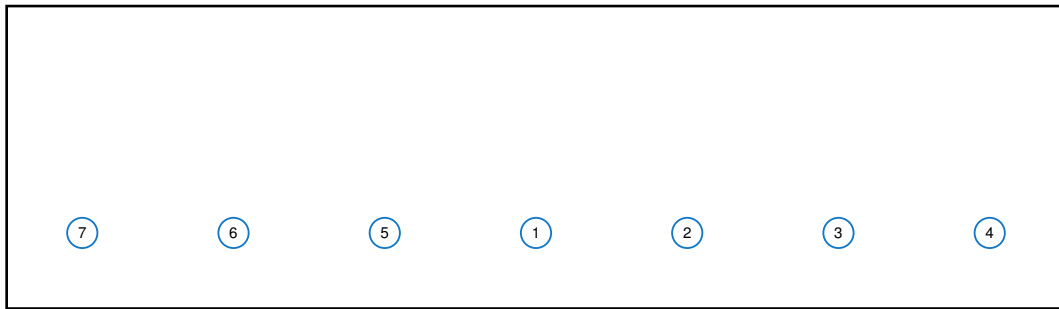
- a)\* Wie lautet die Metrik, die bei der Konstruktion eines SPTs minimiert wird?
- b)\* Wie lautet hingegen die Metrik, die bei der Konstruktion eines Minimum Spanning Tress (MSTs) minimiert wird?

Im Folgenden wollen wir die Unterschiede in der Funktionsweise der Algorithmen von Dijkstra und Bellman-Ford genauer beleuchten. Dazu sei ein Netzwerk in Form der Distanzmatrix

$$D = \begin{bmatrix} 0 & 1 & \infty & \infty & 2 & \infty & \infty \\ 1 & 0 & 1 & \infty & \infty & \infty & \infty \\ \infty & 1 & 0 & 1 & \infty & \infty & \infty \\ \infty & \infty & 1 & 0 & \infty & \infty & 1 \\ 2 & \infty & \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & \infty & \infty & 4 & 0 & 1 \\ \infty & \infty & \infty & 1 & \infty & 1 & 0 \end{bmatrix}$$

gegeben.

- c)\* Zeichnen Sie den Graphen, indem Sie die entsprechenden Kanten und Gewichte in untenstehende Abbildung eintragen.



d)\* Die Anordnung der Knoten im Vordruck von Teilaufgabe c) lässt bereits vermuten, dass die Knoten weitgehend in einer Kette verschaltet sind. Hätte man das auch direkt aus der Distanzmatrix **D** erahnen können? Begründen Sie Ihre Antwort.

**Für die folgenden Teilaufgaben sei 1 der Wurzelknoten.** Falls es in einem Schritt mehr als einen Knoten gibt, mit dem Sie fortfahren können, so wählen Sie stets den Knoten mit der niedrigeren Nummer.

e) Führen Sie auf dem Graphen aus Teilaufgabe c) den Algorithmus von Dijkstra durch. Zeichnen Sie immer den aktuellen Zustand des Netzwerks, wenn ein neuer Knoten in die Menge *S* der bearbeiteten Knoten aufgenommen wird. Notieren Sie außerdem zu jedem Knoten *i* seinen Vorgänger und die derzeitigen Kosten als Tupel  $(p[i], d[i])$ .

f)\* Wieviele Iterationen wird der Algorithmus von Bellman-Ford höchstens benötigen?

g) Führen Sie auf dem Graphen aus Teilaufgabe c) den Algorithmus von Bellman-Ford durch. Zeichnen Sie immer dann den aktuellen Zustand des Netzwerks, nachdem alle Pfade der Länge *N* berücksichtigt worden sind. Notieren Sie außerdem zu jedem Knoten *i* seinen Vorgänger und die derzeitigen Kosten als Tupel  $(p[i], d[i])$ .

### Aufgabe 3 ARP und IP-Fragmentierung

In Abbildung 1 ist eine Anordnung von Netzkomponenten mit ihren IP- und MAC-Adressen dargestellt. Die beiden Computer PC1 und PC2 verwenden den jeweils lokalen Router als Default-Gateway. PC1 sendet ein IP-Paket der Größe *l*=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.

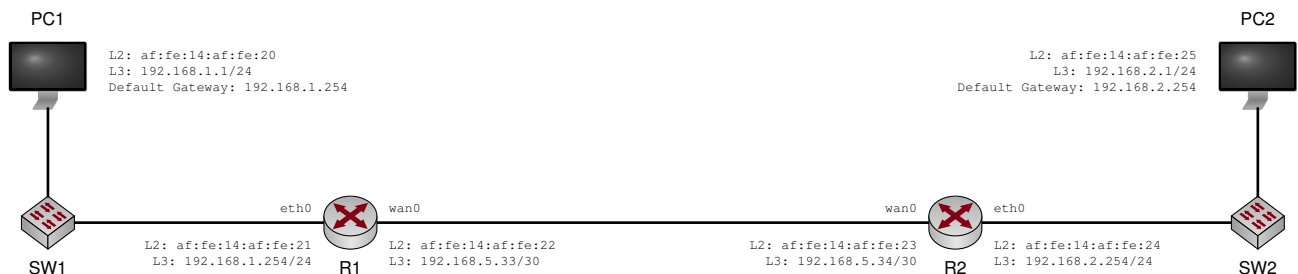


Abbildung 1: Netztopologie

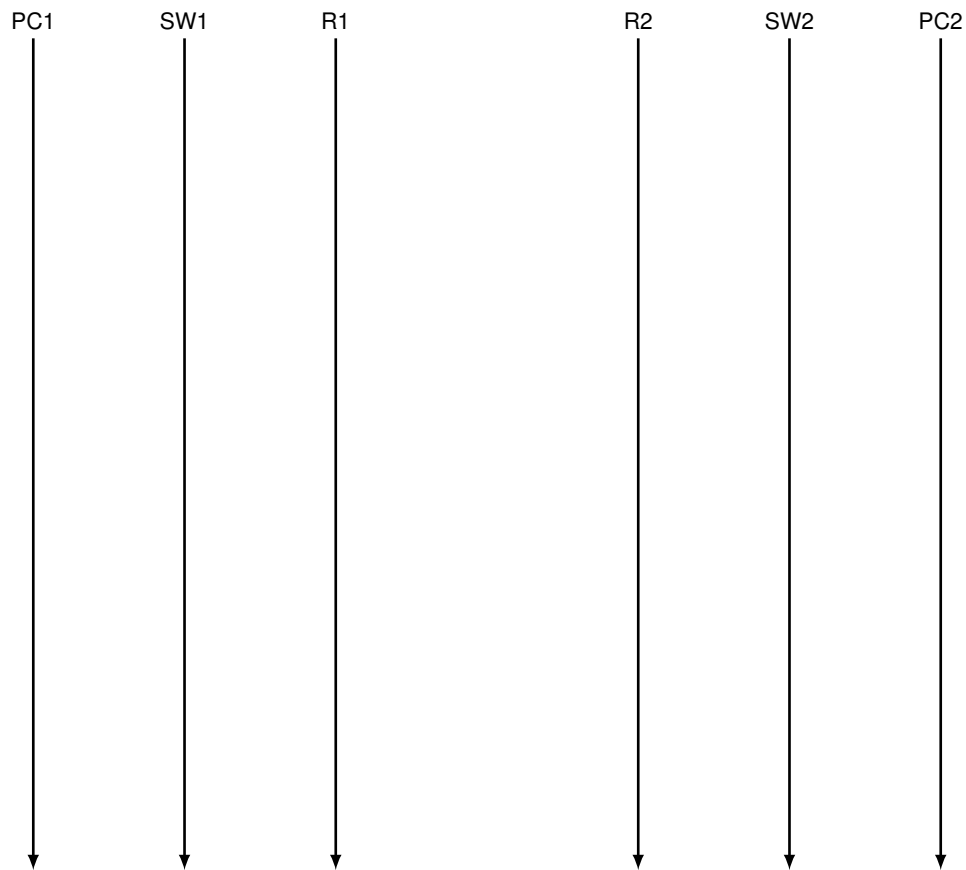
Im Folgenden soll die Übertragung des Pakets mit allen notwendigen Zwischenschritten nachvollzogen werden. Gehen Sie zunächst davon aus, dass die ARP-Caches aller beteiligten Netzwerkkomponenten geleert sind.

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

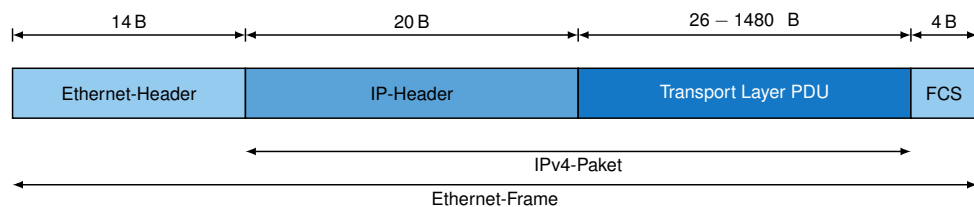
b)\* In wie viele Fragmente muss R1 das Paket von PC1 aufteilen?

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

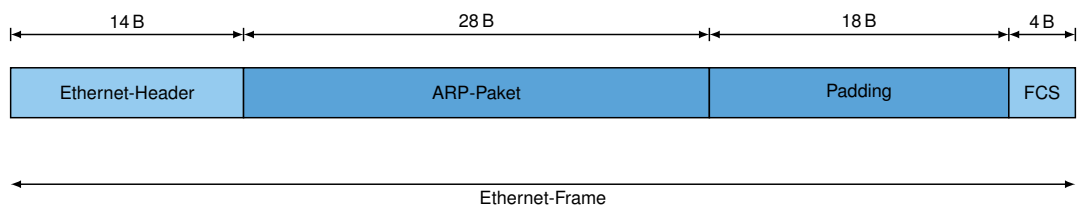
d) Skizzieren Sie ein einfaches Nachrichtenflussdiagramm, welches **alle Rahmen** berücksichtigt, die auf den jeweiligen Verbindungen übertragen werden müssen. **Nennen Sie die Art der ausgetauschten Rahmen und geben Sie den Rahmen Nummern (1,2,3,...).** (Das Diagramm muss nicht Maßstabsgetreu sein, Serialisierungszeiten können vernachlässigt werden.)



Vergegenwärtigen Sie sich den Zusammenhang, dass einem IP-Paket bei seinem Versand ein Ethernet-Header vorangestellt wird. Dieser wird bei jedem Zwischenempfänger (Hop) auf dem Weg zum Zielsystem modifiziert beziehungsweise ersetzt. Der *Payload* eines Ethernet-Frames beinhaltet also insbesondere auch den IP-Header.

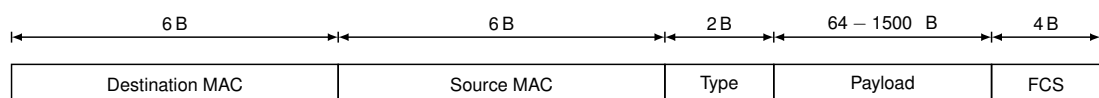


Genauso werden ARP-Requests und -Replies gekapselt.



Um Ihnen das Suchen der entsprechenden Vorlesungsfolien zu ersparen, sind im Folgenden der Ethernet- und IP-Header sowie ARP-Pakete nochmals schematisch dargestellt und kurz erklärt:

#### Ethernet-Header



Das Type-Feld des Ethernet-Headers gibt die Art der Payload an. Die für uns relevanten Werte sind:

- IPv4  $\triangleq$  0x0800
- ARP  $\triangleq$  0x0806

### ARP-Pakete

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
Hardware Type																Protocol Type															
Hardware Addr. Length								Protocol Addr. Length								Operation															
Sender Hardware Address (first 32 bit)																															
Sender Hardware Address (last 16 bit)																Sender Protocol Address (first 16 bit)															
Sender Protocol Address (last 16 bit)																Target Hardware Address (first 16 bit)															
Target Hardware Address (last 32 bit)																															
Target Protocol Address																															

Die Felder nehmen dabei folgende Werte an:

- Hardware Address Type für Ethernet-Adressen ist 0x0001
- Protocol Address Type für IPv4 ist 0x0800
- Hardware bzw. Protocol Address Length geben die Länge der jeweiligen Adresse in Byte an
- Opcode ist 0x0001 (ARP-Request) oder 0x0002 (ARP-Reply)

### IP-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			
Version				IHL				TOS								Total Length																		
Identification																R	D F	M F	Fragment Offset															
TTL								Protocol								Header Checksum																		
Source Address																																		
Destination Address																																		
Options / Padding (optional)																																		
Payload																																		

Auch hier wieder einige Hinweise zu den Feldern:

- „Version“ ist für IPv4 stets 0x4
- „IHL“ (IP Header Length) gibt die Länge des IP-Headers in ganzzahligen Vielfachen von 4 B an
- „Total Length“ gibt die Gesamtlänge des IP-Pakets inkl. Header in Vielfachen von 1 B an
- „Identification“ ist ein eindeutiger 16 bit langer Wert, welcher zum Reassemblieren von Fragmenten benötigt wird
- Die drei Flags „R“, „DF“ und „MF“ stehen für „Reserved“, „Do not Fragment“ und „More Fragments“. Ersteres hat also keine Bedeutung, die beiden letzteren geben an, ob das Paket fragmentiert werden darf (0 falls ja) und ob mehr Fragmente folgen (1 falls ja).
- „Fragment Offset“ gibt an, an welche Position die in der Payload enthaltenen Daten beim Reassemblieren geschrieben werden sollen. Der Offset wird in ganzzahligen Vielfachen von 8 B angegeben!
- Die TTL wird pro Hop (also Weiterleitung durch einen Router) um 1 dekrementiert. Erreicht sie 0, wird das Paket verworfen und ein ICMP TTL Exceeded an den Sender zurückgeschickt. Der Initialwert der TTL bleibt dem Sender überlassen – ein gebräuchlicher Wert ist 64.

**Am Ende dieses Übungsblatts finden Sie Vordrucke für Ethernet-Header, ARP-Pakete (Header und Payload) 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<sub>(10)</sub> für dezimal.

e) Füllen Sie für die ersten drei Rahmen aus Teilaufgabe (d) jeweils einen Ethernet-Header und die passende Payload (ARP-Paket oder IP-Header mit angedeuteter Payload) aus. Beschriften Sie die gestrichelte Box neben dem jeweiligen Header/Paket mit der in (d) vergebenen Rahmennummer.

f) Füllen Sie für alle übrigen Rahmen, welche eine IP-Payload transportieren, jeweils einen Ethernet- und IP-Header aus. Beschriften Sie die gestrichelte Box neben dem jeweiligen Header mit der in (d) vergebenen Rahmennummer.

## Vordrucke für Protokoll-Header:

### Ethernet Frames

<input type="checkbox"/>				Payload	FCS
<input type="checkbox"/>				Payload	FCS
<input type="checkbox"/>				Payload	FCS
<input type="checkbox"/>				Payload	FCS
<input type="checkbox"/>				Payload	FCS
<input type="checkbox"/>				Payload	FCS
<input type="checkbox"/>				Payload	FCS

### ARP-Pakete

<input type="checkbox"/>	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

<input type="checkbox"/>	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



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

IP-Pakete



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



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



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

