

# GRNVS Tutorium 07 - SS21

Fabian Sauter

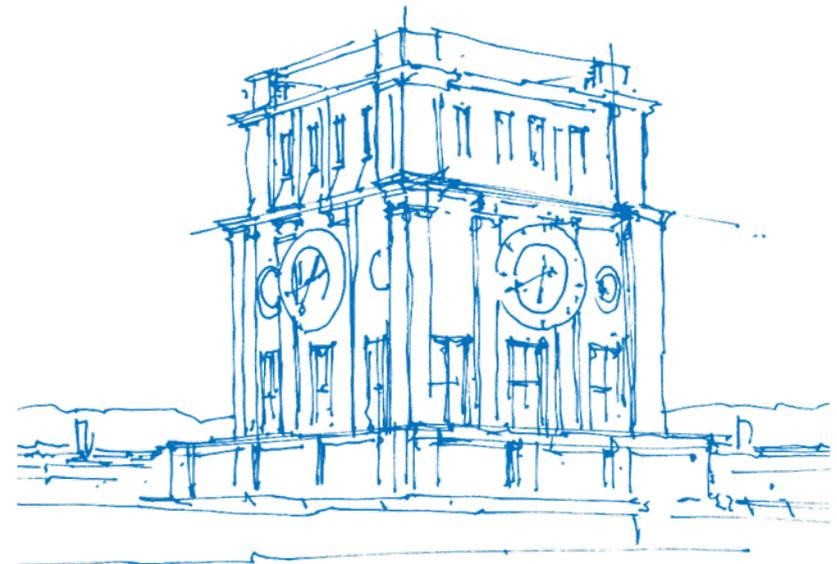
Technische Universität München

Grundlagen: Rechnernetze und Verteilte Systeme (IN0010)

Lehrstuhl für Netzarchitekturen und Netzdienste

Garching, 07.06.2021

**Slides & Notes:** <http://grnvs.uwpx.org>

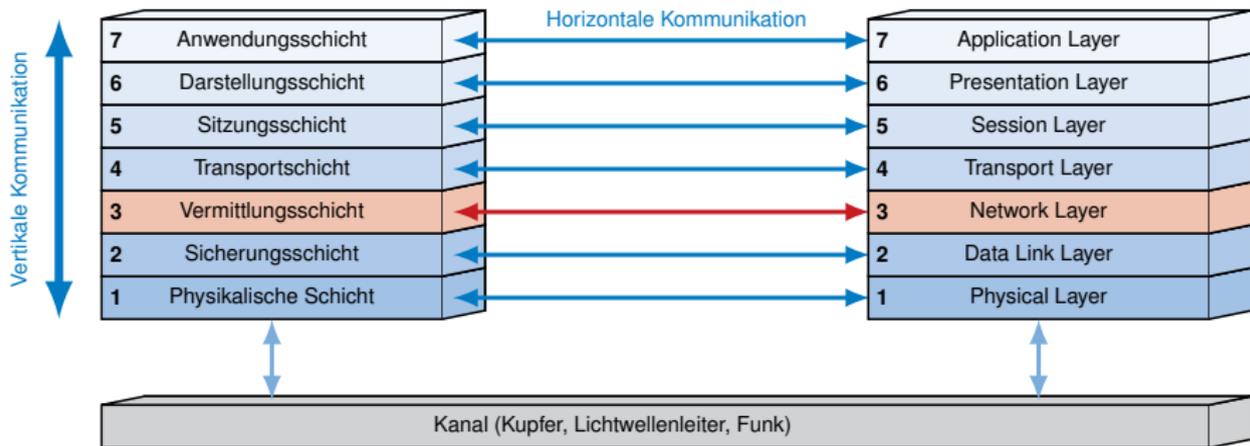


*TUM Uhrenturm*

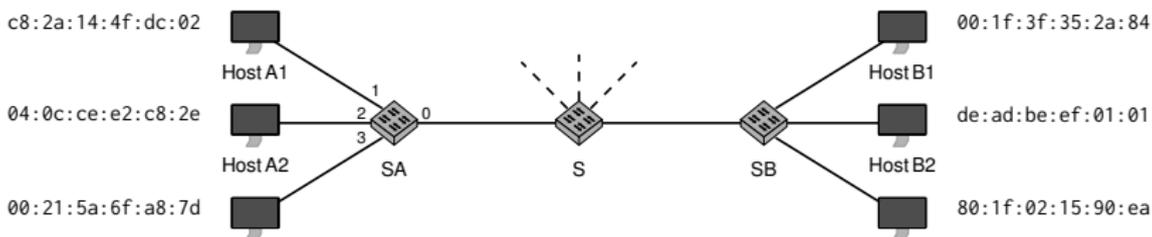
# Wiederholung

# Kapitel 3: Vermittlungsschicht

## Einordnung im ISO/OSI-Modell



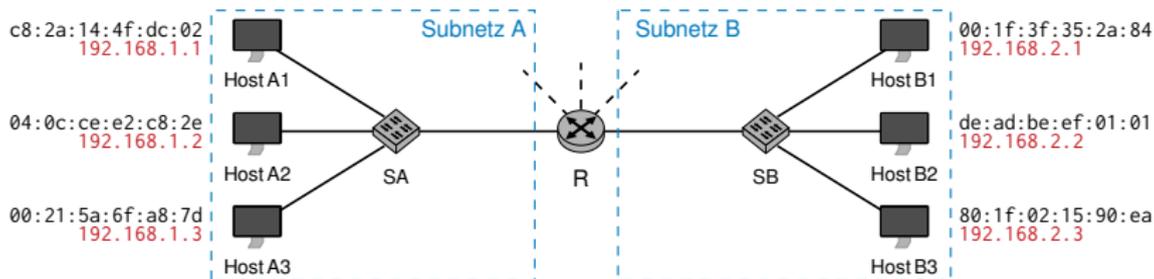
Wir betrachten das folgende Beispielnetz, welches auf einem aktuellen Ethernet-Standard basiert:



Wie viele Einträge enthält die Switching-Tabelle von Switch SA?

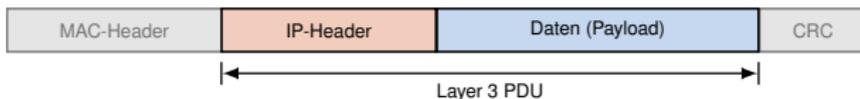
## Internet Protocol version 4 (IPv4) [14]

Wir betrachten das Beispielnetz mit einem Router (R) in der Mitte:



- Jedem Host ist eine **IP-Adresse** zugewiesen.
- Jede IP-Adresse ist in vier Gruppen zu je einem Byte, durch Punkte getrennt, dargestellt (**Dotted Decimal Notation**).
- In diesem Beispiel identifiziert das 4. Oktett einen Host innerhalb eines Netzes.
- Die ersten drei Oktette identifizieren das Netzwerk, in dem sich der Host befindet.
- Der Router R trifft Weiterleitungsentscheidungen auf Basis der Ziel-IP-Adresse.

⇒ Jedes Paket muss mit einer Absender- und Ziel-IP-Adresse (im IP-Header) versehen werden:



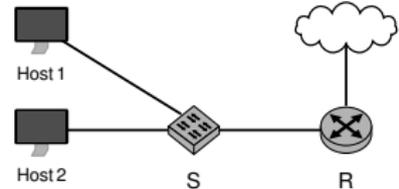
## Internet Protocol version 4 (IPv4) [14]

### Adressauflösung [15]

- Host 1 will eine Nachricht an Host 2 senden
- Die IP-Adresse von Host 2 (192.168.1.2) sei ihm bereits bekannt
- Wie erhält Host1 die zugehörige MAC-Adresse?

c8:2a:14:4f:dc:02  
192.168.1.1

04:0c:ce:e2:c8:2e  
192.168.1.2



### Address Resolution Protocol (ARP)

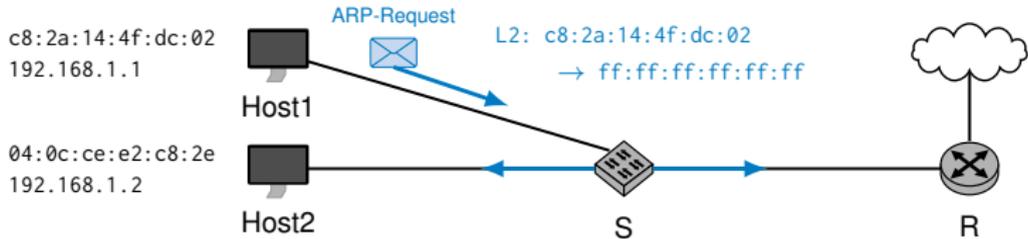
1. Host1 sendet einen ARP Request: „Who has 192.168.1.2? Tell 192.168.1.1 at c8:2a:14:4f:dc:02“
2. Host2 antwortet mit einem ARP Reply: „192.168.1.2 is at 04:0c:ce:e2:c8:2e“

Offset	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
0B	Hardware Type																Protocol Type															
4B	Hardware Addr. Length								Protocol Addr. Length								Operation															
8B	Sender Hardware Address (first 32 bit)																															
12B	Sender Hardware Address (last 16 bit)																Sender Protocol Address (first 16 bit)															
16B	Sender Protocol Address (last 16 bit)																Target Hardware Address (first 16 bit)															
20B	Target Hardware Address (last 32 bit)																															
24B	Target Protocol Address																															

Abbildung 2: ARP-Paket für IPv4 über Ethernet

# Internet Protocol version 4 (IPv4) [14]

## Beispiel:



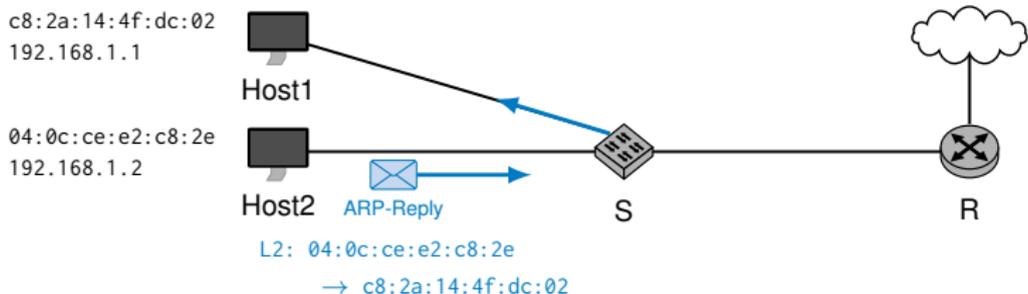
**Hinweis:** L2: xx:xx:xx:xx:xx:xx → yy:yy:yy:yy:yy:yy stellt Quell- und Ziel-MAC-Adresse auf Schicht 2 dar.

Offset	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
0B	0x0001 (Ethernet)												0x8000 (IPv4)																			
4B	0x06				0x04				0x0001 (Request)																							
8B	0xc82a144f																															
12B	0xc802 (Sender Hardware Address)												0xc8a8																			
16B	0x0101 (Sender Protocol Address)												0x0000																			
20B	0x00000000 (Target Hardware Address)																															
24B	0xc8a8102 (Target Protocol Address)																															

(a) ARP Request

- Der ARP-Request wird an die MAC-Broadcast-Adresse ff:ff:ff:ff:ff:ff geschickt, weswegen der Switch S den Rahmen an alle angeschlossenen Hosts weiterleitet.

## Beispiel:



**Hinweis:** L2: xx:xx:xx:xx:xx:xx → yy:yy:yy:yy:yy:yy stellt Quell- und Ziel-MAC-Adresse auf Schicht 2 dar.

Offset	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
0B	0x0001 (Ethernet)												0x8000 (IPv4)																			
4B	0x06				0x04				0x0001 (Request)																							
8B	0xc82a144f																															
12B	0xdc02 (Sender Hardware Address)												0xc8a8																			
16B	0x101 (Sender Protocol Address)												0x0000																			
20B	0x00000000 (Target Hardware Address)																															
24B	0xc8a80102 (Target Protocol Address)																															

(c) ARP Request

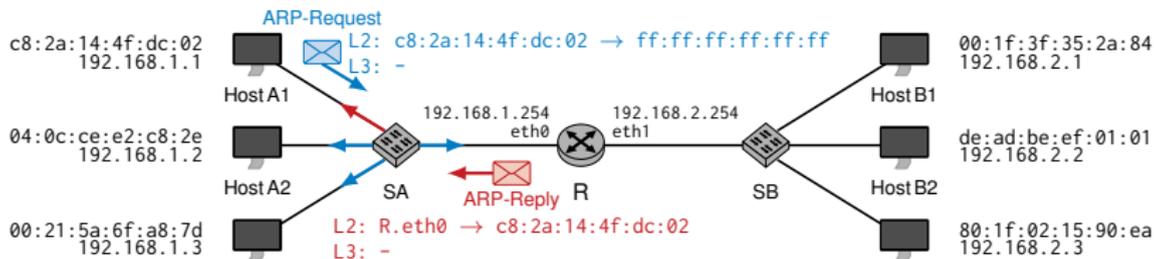
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0B	0x0001 (Ethernet)												0x8000 (IPv4)																			
4B	0x06				0x04				0x0002 (Reply)																							
8B	0x040ccee2																															
12B	0xc82e												0xc8a8																			
16B	0x0102												0xc82a																			
20B	0x144fdc02																															
24B	0xc8a80101																															

(d) ARP Reply

- Der ARP-Request wird an die MAC-Broadcast-Adresse ff:ff:ff:ff:ff:ff geschickt, weswegen der Switch S den Rahmen an alle angeschlossenen Hosts weiterleitet.
- Der ARP-Reply wird als MAC-Unicast versendet (adressiert an Host1).
- Die Rollen Sender/Target sind zwischen Request und Reply vertauscht (vgl. Inhalte der grünen und roten Felder).

Was ist nun, wenn das Ziel **nicht** im selben Netz liegt (z. B. Host A1 an Host B2)?

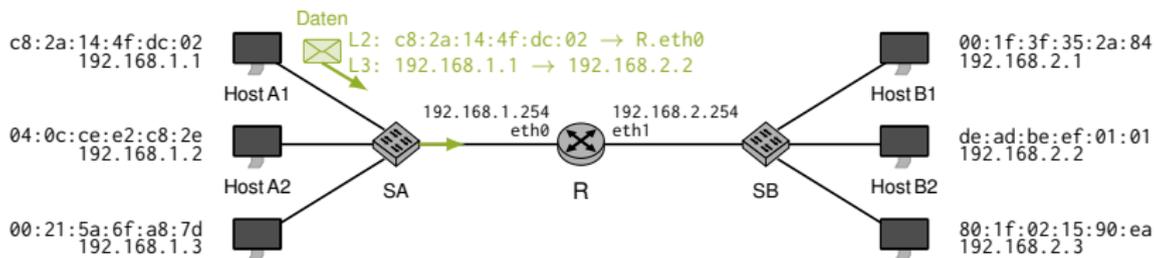
- Jeder Host sollte einen Router zum Internet, das sog. **Default Gateway**, kennen, an das er alle Pakete schickt, deren Zieladressen nicht im eigenen Netz liegen, und für die in seiner Routing-Tabelle nicht ein spezifisches Gateway eingetragen ist.
- Ob eine Zieladresse zum eigenen Netz gehört erkennt ein Host durch Vergleich der Zieladresse mit der eigenen Netzadresse.
- Im Moment gehen wir noch davon aus, dass die ersten 3 Oktette einer IP-Adresse das Netz identifizieren  
 ⇒ 192.168.1.1 und 192.168.2.2 liegen in unterschiedlichen Netzen.



1. Host A1 erkennt, dass 192.168.2.2 nicht im eigenen Netz liegt. Sein Default-Gateway ist 192.168.1.254.
2. Host A1 löst die MAC-Adresse zu 192.168.1.254 auf.

Was ist nun, wenn das Ziel **nicht** im selben Netz liegt (z. B. Host A1 an Host B2)?

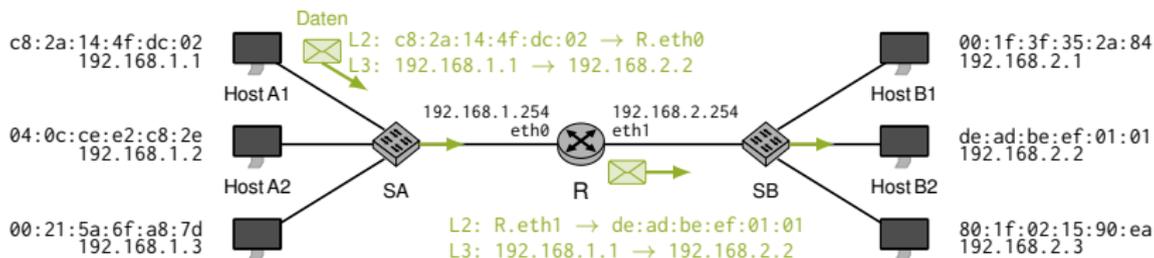
- Jeder Host sollte einen Router zum Internet, das sog. **Default Gateway**, kennen, an das er alle Pakete schickt, deren Zieladressen nicht im eigenen Netz liegen, und für die in seiner Routing-Tabelle nicht ein spezifisches Gateway eingetragen ist.
- Ob eine Zieladresse zum eigenen Netz gehört erkennt ein Host durch Vergleich der Zieladresse mit der eigenen Netzadresse.
- Im Moment gehen wir noch davon aus, dass die ersten 3 Oktette einer IP-Adresse das Netz identifizieren  
 ⇒ 192.168.1.1 und 192.168.2.2 liegen in unterschiedlichen Netzen.



1. Host A1 erkennt, dass 192.168.2.2 nicht im eigenen Netz liegt. Sein Default-Gateway ist 192.168.1.254.
2. Host A1 löst die MAC-Adresse zu 192.168.1.254 auf.
3. Host A1 sendet das Datenpaket an R: Dabei adressiert er R mittels der eben bestimmten MAC-Adresse (Schicht 2). Als Ziel-IP-Adresse (Schicht 3) verwendet er die IP-Adresse von Host B2.

Was ist nun, wenn das Ziel **nicht** im selben Netz liegt (z. B. Host A1 an Host B2)?

- Jeder Host sollte einen Router zum Internet, das sog. **Default Gateway**, kennen, an das er alle Pakete schickt, deren Zieladressen nicht im eigenen Netz liegen, und für die in seiner Routing-Tabelle nicht ein spezifisches Gateway eingetragen ist.
- Ob eine Zieladresse zum eigenen Netz gehört erkennt ein Host durch Vergleich der Zieladresse mit der eigenen Netzadresse.
- Im Moment gehen wir noch davon aus, dass die ersten 3 Oktette einer IP-Adresse das Netz identifizieren  
 ⇒ 192.168.1.1 und 192.168.2.2 liegen in unterschiedlichen Netzen.



1. Host A1 erkennt, dass 192.168.2.2 nicht im eigenen Netz liegt. Sein Default-Gateway ist 192.168.1.254.
2. Host A1 löst die MAC-Adresse zu 192.168.1.254 auf.
3. Host A1 sendet das Datenpaket an R: Dabei adressiert er R mittels der eben bestimmten MAC-Adresse (Schicht 2). Als Ziel-IP-Adresse (Schicht 3) verwendet er die IP-Adresse von Host B2.
4. R akzeptiert das Paket, bestimmt das ausgehende Interface und leitet das Paket weiter an Host B2. Dabei adressiert R wiederum Host B2 anhand seiner MAC-Adresse (erfordert ggf. einen weiteren ARP-Schritt).

## Internet Protocol version 4 (IPv4) [14]

### Subnetting (Classless Routing)

Bereits 1993 wurde mit CIDR<sup>5</sup> ein Verfahren zur Unterteilung von IP-Netzen eingeführt:

- Zusätzlich zur IP-Adresse erhält ein Interface eine ebenfalls 32 bit lange **Subnetzmaske**
- Die Subnetzmaske unterteilt die IP-Adresse in einen **Netzanteil** und einen **Hostanteil**
- Eine logische 1 in der Subnetzmaske bedeutet Netzanteil, eine logische 0 Hostanteil
- UND-Verknüpfung von IP-Adresse und Subnetzmaske ergibt die Netzadresse
- Die übliche Klassenzugehörigkeit hat damit nur noch im Sprachgebrauch eine Bedeutung

#### Beispiel 1:

IP-Adresse	11000000 . 10101000 . 00000000 . 10110010	192.168.0.178
Subnetz Maske	11111111 . 11111111 . 11111111 . 00000000	255.255.255.0
Netzadresse	11000000 . 10101000 . 00000000 . 00000000	192.168.0.0
Broadcastadresse	11000000 . 10101000 . 00000000 . 11111111	192.168.0.255

- 24 bit Netzanteil, 8 bit Hostanteil  $\Rightarrow 2^8 = 256$  Adressen
- Netzadresse: 192.168.0.0
- Broadcast-Adresse: 192.168.0.255
- Nutzbare Adressen für Hosts:  $2^8 - 2 = 254$

<sup>5</sup> Classless Inter-Domain Routing

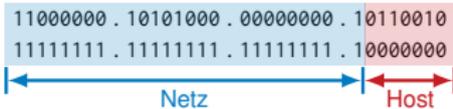
## Internet Protocol version 4 (IPv4) [14]

### Subnetting (Classless Routing)

Bereits 1993 wurde mit **CIDR**<sup>5</sup> ein Verfahren zur Unterteilung von IP-Netzen eingeführt:

- Zusätzlich zur IP-Adresse erhält ein Interface eine ebenfalls 32 bit lange **Subnetzmaske**
- Die Subnetzmaske unterteilt die IP-Adresse in einen **Netzanteil** und einen **Hostanteil**
- Eine logische 1 in der Subnetzmaske bedeutet Netzanteil, eine logische 0 Hostanteil
- UND-Verknüpfung von IP-Adresse und Subnetzmaske ergibt die Netzadresse
- Die übliche Klassenzugehörigkeit hat damit nur noch im Sprachgebrauch eine Bedeutung

#### Beispiel 2:

IP-Adresse	11000000 . 10101000 . 00000000 . 10110010	192.168.0.178
Subnetz Maske	11111111 . 11111111 . 11111111 . 10000000	255.255.255.128
		
Netzadresse	11000000 . 10101000 . 00000000 . 10000000	192.168.0.128
Broadcastadresse	11000000 . 10101000 . 00000000 . 11111111	192.168.0.255

- 25 bit Netzanteil, 7 bit Hostanteil  $\Rightarrow 2^7 = 128$  Adressen
- Netzadresse: 192.168.0.128
- Broadcast-Adresse: 192.168.0.255
- Nutzbare Adressen für Hosts:  $2^7 - 2 = 126$

<sup>5</sup> Classless Inter-Domain Routing

## Internet Protocol version 4 (IPv4) [14]

### Subnetting (Classless Routing)

Bereits 1993 wurde mit **CIDR**<sup>5</sup> ein Verfahren zur Unterteilung von IP-Netzen eingeführt:

- Zusätzlich zur IP-Adresse erhält ein Interface eine ebenfalls 32 bit lange **Subnetzmaske**
- Die Subnetzmaske unterteilt die IP-Adresse in einen **Netzanteil** und einen **Hostanteil**
- Eine logische 1 in der Subnetzmaske bedeutet Netzanteil, eine logische 0 Hostanteil
- UND-Verknüpfung von IP-Adresse und Subnetzmaske ergibt die Netzadresse
- Die übliche Klassenzugehörigkeit hat damit nur noch im Sprachgebrauch eine Bedeutung

#### Beispiel 3:

IP-Adresse	11000000 . 10101000 . 00000000 . 10110010	192.168.0.178
Subnetz Maske	11111111 . 11111111 . 11111111 . 11000000	255.255.255.192
Netzadresse	11000000 . 10101000 . 00000000 . 10000000	192.168.0.128
Broadcastadresse	11000000 . 10101000 . 00000000 . 10111111	192.168.0.191

- 26 bit Netzanteil, 6 bit Hostanteil  $\Rightarrow 2^6 = 64$  Adressen
- Netzadresse: 192.168.0.128
- Broadcast-Adresse: 192.168.0.191
- Nutzbare Adressen für Hosts:  $2^6 - 2 = 62$

<sup>5</sup> Classless Inter-Domain Routing

# Ablauf:

- Aufgabe 1
- Aufgabe 2

# Studenten zählen

(Nur als Erinnerung für mich.)