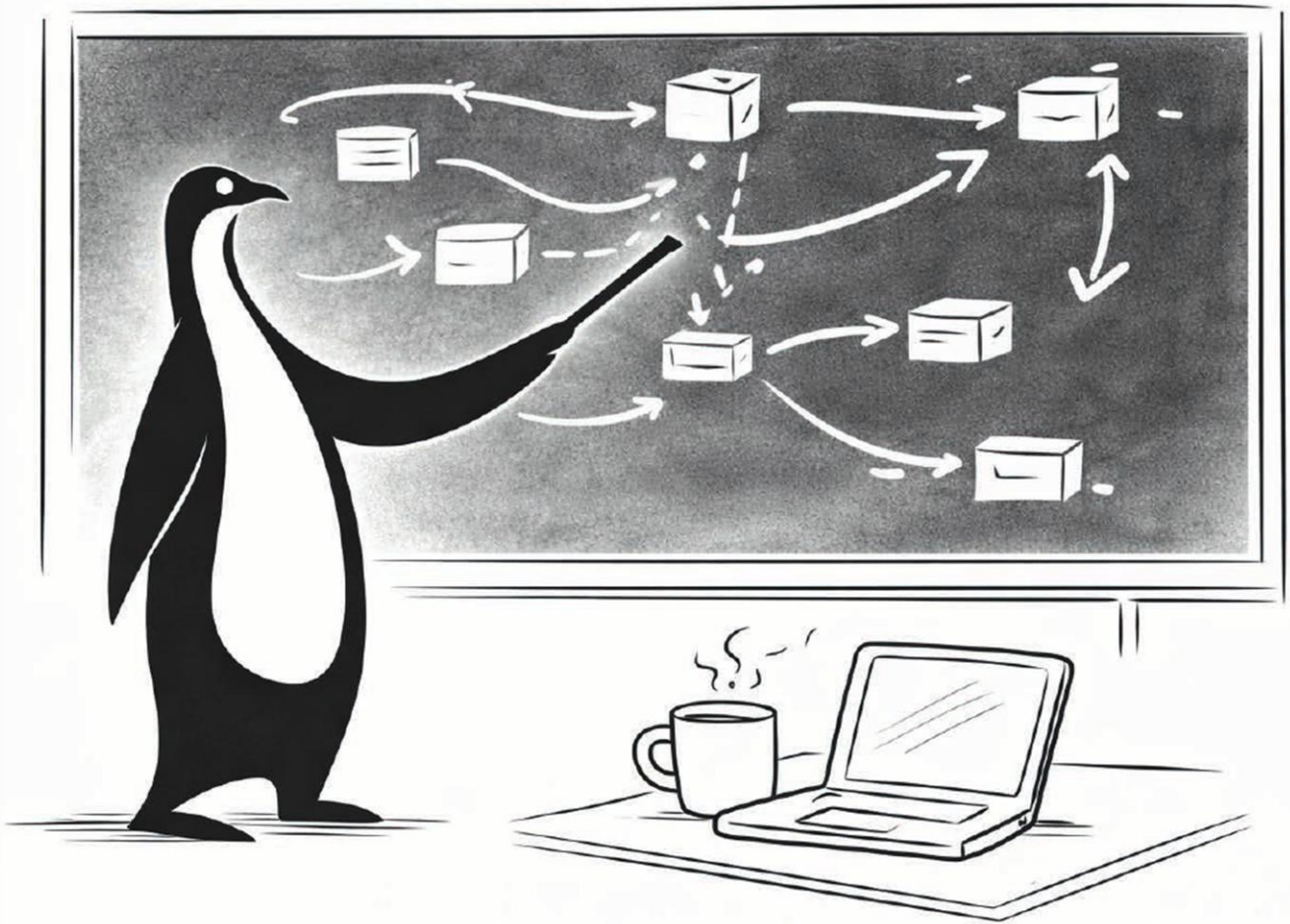


# Netzwerktechnik-Kochbuch

Ein roter Faden durch das OSI-Menü



Ulf Treue

**Gelingt es,  
Menschen für ein gemeinsames Ziel zu motivieren,  
alle Ressourcen optimal zu nutzen und  
unterwegs auf Augenhöhe zu bleiben,  
kann Großes entstehen.**

Für die Unterstützung gilt mein Dank (in chronologischer Folge):

Kerstin  
Herrn Henschke  
Herrn Michaelis  
Herrn Wunderlich  
Herrn Schlegel  
Frau Lucas  
Frau Kalenteva  
Herrn Dr. Riesener

Ein besonderer Dank gilt unserer Therapie-Katze Miri,  
die stets ein wachsames Auge auf mich hatte:



**Sehr geehrte Damen und Herren,  
ich begrüße Sie zu Ihrem Unterricht im Fach Netzwerktechnik.**

Dieses "Kochbuch" ist entstanden, um Ihnen ein Werkzeug an die Hand zu geben, mit dem Sie sich einen Weg durch den Dschungel der Netzwerktechnik bahnen können. Es ist nicht als ein herkömmliches Lehrbuch zu verstehen, vielmehr sollte es als „Roter Faden“ betrachtet werden.

- Das Symbol "=>" bedeutet: "daraus folgt" oder auch "in diese Richtung weiterdenken"
- Das Symbol "==>>" bedeutet: "Denk daran!" oder auch "Wichtig!"
- Die runden Klammern ( ) geben noch zusätzliche Hinweise.

Orthographie ist mir kein Fremdwort, aber:

Wer Fehler im Kochbuch findet (daran dürfte es nicht mangeln), darf sie behalten ;-).  
Nein, nein. Bitte geben Sie mir eine Rückmeldung, nur so kann dieses "Werk" besser werden.

Die Grafiken sind in einer Symbolik dargestellt, die an Cisco erinnert.

Auch in Filius ist diese Symbolik verfügbar.

==>> Ein Router wird RUND dargestellt, der Switch ist eckig.

==>> Der Inhalt des Kochbuches steht unter der Lizenz: CC BY-NC-ND!

**Ich wünsche Ihnen viel Spaß und viel Ausdauer!**

# Inhaltsverzeichnis

0	Quellen.....	1
0.1	Quellen aus dem Internet.....	1
0.2	Lernsoftware-filius.de.....	1
0.2.1	Simulationen von Filius.....	1
1	Allgemein.....	2
1.1	Geschichte.....	2
1.2	Leistungsvermittlung, Paketvermittlung.....	2
1.3	Kommunikationsprotokoll.....	2
1.4	Netzwerk-Nodes.....	2
1.5	Vorteile von Netzwerken.....	2
1.6	Nachteile von Netzwerken.....	2
1.7	Dimensionen von Netzwerken.....	3
1.8	Topologien (allgemein).....	3
1.9	Echtzeitfähigkeit.....	3
1.10	Richtungsunabhängigkeit.....	3
2	OSI-Schichtenmodell.....	4
2.1	Warum Schichten.....	4
2.2	verschiedene Schichtenmodelle.....	4
2.3	ISO-OSI-7-Schichten-Modell.....	4
2.4	TCP/IP-Modell.....	4
2.5	TCP/IP-Modell und Schokolade.....	4
3	Schicht Null.....	6
3.1	Koaxialkabel.....	6
3.2	Twisted-Pair-Kabel.....	6
3.2.1	CAT 3.....	6
3.2.2	CAT 5.....	6
3.2.3	CAT 5e.....	6
3.2.4	CAT 6.....	6
3.2.5	CAT 6 <sub>A</sub> .....	7
3.2.6	CAT 7 / 8.....	7
3.3	Lichtwellenleiter (LWL).....	7
3.3.1	Multimode.....	7
3.3.2	Singlemode.....	7
3.4	Dämpfung (speziell LWL).....	7
3.4.1	Welches Licht können wir sehen?.....	7
3.4.2	Zuordnung in der Praxis.....	8
3.4.3	Geschwindigkeiten und Entfernungen von LWL.....	8
3.5	Strukturierte Verkabelung.....	8
3.5.1	Bindeglied zwischen TP-Kabel und LWL.....	8
3.6	Einfache Regel für die Verkabelung.....	9
3.7	Wireless LAN (WLAN).....	9

3.7.1	Normen von WLAN und deren (Vermarktungs-Name).....	9
3.7.2	MIMO / MU-MIMO .....	9
3.7.2.1	MIMO.....	10
3.7.2.2	MU-MIMO .....	10
3.7.3	Sicherheit im WLAN (kurz und knapp).....	10
3.8	Dezibel [dB].....	10
3.9	Antennengewinn.....	11
3.9.1	Beispiel: Taschenlampe .....	11
3.9.2	EIRP, ERP „das Blenden“ .....	11
4	Ethernet-Frame .....	12
4.1	Präambel.....	12
4.2	"Rahmenformate Ethernet II" .....	12
4.3	CRC/FCS .....	12
4.4	Problem der Größe des Ethernet-Frames.....	12
4.4.1	Jumbo-Frame .....	12
5	Sniffer .....	13
5.1	Rechtliches.....	13
5.2	Einfacher Sniffer.....	13
5.3	Wireshark.....	13
5.4	Netzwerkkarte "Freizügiger Modus" .....	13
5.5	Zu empfehlende Konsolen-Sniffer: .....	13
5.5.1	tcpdump (Linux).....	13
5.5.2	Windump .....	13
6	Schicht Eins.....	14
6.1	Netzwerkkarte .....	14
6.1.1	Hauptaufgaben der Netzwerkkarte .....	14
6.2	Netzzugriffsverfahren .....	14
6.2.1	Kollisionserkennung (CSMA/CD).....	14
6.2.2	Kollisionsvermeidung (CSMA/CA) .....	14
6.3	HUB: .....	15
7	Schicht Zwei.....	16
7.1	MAC-Adresse.....	16
7.2	Switch .....	16
7.2.1	Wie wird die SAT-Tabelle aufgebaut? .....	16
7.2.2	Unmanaged Switch (Desktop Switch).....	16
7.2.3	Managed Switch.....	17
7.2.4	Features moderner (managed) Switches:.....	17
7.2.4.1	Port-Mirroring.....	17
7.2.4.2	Link Aggregation .....	17
7.2.4.3	Power over Ethernet .....	18
7.2.4.4	Spanning Tree .....	21
7.2.4.5	VLAN .....	22
7.2.4.6	Stackable Switch.....	23

8	Schicht Drei.....	24
8.1	IPv4	24
8.1.1	IP-Adresse .....	24
8.1.2	Subnetzmaske.....	25
8.1.3	IPv4 Klassen (Class).....	25
8.1.3.1	Class A .....	26
8.1.3.2	Class B .....	26
8.1.3.3	Class C.....	26
8.1.4	Besonderheiten .....	27
8.1.5	Besondere IPv4 Adressen (Auszug).....	28
8.1.5.1	Class A .....	28
8.1.5.2	Class B .....	28
8.1.5.3	Class C .....	28
8.1.6	Standardsubnetzmaske vs. Subnetzmaske .....	29
8.1.6.1	Problem 1 .....	29
8.1.6.2	Problem 2 .....	29
8.1.7	Subnetting IPv4 ("Richtiges Subnetting").....	30
8.1.7.1	Subnetting in 2 Subnetze .....	31
8.1.7.2	Subnetting in 4 Subnetze .....	33
8.1.7.3	Subnetting in 8 Subnetze .....	35
8.1.7.4	Asymmetrisches Subnetting (VLSM).....	36
8.1.7.5	Subnetting Class A und Class B.....	36
8.1.8	"Reverses Subnetting" IPv4.....	37
8.1.9	Subnetzmaske IPv4 analysieren.....	39
8.2	IPv6	41
8.2.1	„Kürzen“ von IPv6-Adressen.....	42
8.2.2	„Weiter verkürzen“ von IPv6-Adressen .....	43
8.2.3	Subnetting IPv6 .....	45
8.2.3.1	Subnetting in 2 Subnetze .....	46
8.2.3.2	Subnetting in 4 Subnetze .....	47
8.2.4	"Reverses Subnetting" IPv6.....	49
8.2.5	Besondere IPv6-Adressen (Auszug).....	51
8.2.5.1	Nicht spezifizierte IPv6-Adresse.....	51
8.2.5.2	localhost / loopback-Adresse .....	51
8.2.5.3	Global Unicast .....	51
8.2.5.4	Unique Local Unicast.....	51
8.2.5.5	Link Local Unicast.....	52
8.2.5.6	Multicast .....	52
8.2.6	SLAAC - Stateless Address Autoconfiguration.....	53
8.3	Vergleich der Header von IPv4 und IPv6 .....	54
8.3.1	IPv4-Header .....	54
8.3.2	IPv6 Header .....	54
8.4	DHCP55	
8.4.1	Allgemeine Erklärung zum Ablauf .....	56
8.4.2	Tipps zu DHCP.....	56
8.5	Namensauflösung .....	57
8.5.1	Netbios-Name .....	57

8.5.2	DNS-Name (FQDN).....	58
8.5.2.1	Die Struktur von DNS.....	58
8.5.2.2	Praktische Lösung für Zugriff aus dem Internet.....	59
8.5.2.3	Praktische Lösung für interne Namensauflösung.....	59
8.6	Routing.....	60
8.6.1	Dynamisches Routing im LAN (Navi im Auto).....	60
8.6.2	Statisches Routing ("Ich fahre immer dort entlang").....	61
8.6.2.1	"Net-Routing" 192.168.2.0 (Syntax Windows).....	62
8.6.2.2	"Host-Routing" 192.168.2.22 (Syntax Windows).....	62
8.7	Layer-3-Switch.....	63
9	Schicht Vier.....	64
9.1	Ports 64	
9.1.1	Warum Ports?.....	65
9.1.2	Schreibweisen von IP-Adresse und Port.....	67
9.1.3	Aufteilung der Ports.....	67
9.1.3.1	"System Ports".....	67
9.1.3.2	"User Ports".....	68
9.1.3.3	"Dynamic/Private Ports".....	68
9.2	Protokolle der Schicht 4.....	69
9.2.1	UDP.....	69
9.2.2	TCP.....	69
9.2.2.1	VerbindungsAUFbau zwischen Client und Server.....	70
9.2.2.2	Die kontrollierte Datenübertragung bei TCP.....	71
9.2.2.3	VerbindungsABbau zwischen Client und Server.....	71
9.2.2.4	TCP-Header.....	72
9.2.3	QUIC.....	72
9.3	Portknocking.....	72
9.3.1	Zielsetzung.....	73
9.3.2	Beschreibung.....	73
9.4	Portforwarding / Destination NAT.....	74
9.4.1	Zielsetzung.....	74
9.4.2	Begriffe.....	74
9.4.3	Beschreibung.....	75
9.4.4	Anmerkungen.....	75
9.5	NAT (PAT) / Source NAT.....	76
9.5.1	Zielsetzung.....	76
9.5.2	Begriffe.....	76
9.5.3	Beschreibung.....	77
9.5.4	Ablauf (step by step) für den PC.....	77
9.6	Black- und Whitelist.....	78
9.6.1	Allowlist.....	78
9.6.2	Blocklist.....	79
9.6.3	Kombination aus Allowlist UND Blocklist.....	80
9.6.4	Squidguard.....	81
10	Firewalls.....	82
10.1	Unterscheidungsmerkmale.....	83

10.2 Steuerbare OSI-Schichten der SPI-FW unter Linux .....	84
10.3 Firewall als Brücke zwischen einer Insel und dem Festland .....	85
10.4 Innerer Aufbau der SPI-FW mit iptables .....	86
10.5 Beispiel einer einfachen Firewall für den praktischen Einstieg .....	88
10.5.1 Iptables-Syntax betrachten .....	90
10.5.2 Umsetzung in der Praxis .....	91
10.6 Demilitarisierte Zone ("DMZ") .....	92
10.6.1 Allgemein .....	92
10.6.2 Das 2-stufige Konzept .....	92
10.6.3 Das 1-stufige Konzept .....	92
10.7 Beispiel einer komplexeren Firewall mit einer DMZ .....	93
10.7.1 Auch hier wieder etwas „reverse engineering“ .....	94
11 Verschlüsselung .....	95
11.1 Unterscheidungsmerkmale .....	95
11.2 Symmetrisches Verfahren .....	96
11.2.1 Symmetrische Verschlüsselung in der Übersicht .....	97
11.2.2 Verschlüsselung von Alice zu Bob .....	97
11.2.3 Verschlüsselung von Bob zu Alice .....	97
11.2.4 Das Problem der Übergabe des symmetrischen Schlüssels .....	98
11.3 Asymmetrisches Verfahren .....	99
11.3.1 Beispiel einer selbst gebauten Mechanik .....	99
11.3.2 Asymmetrisches Verfahren am Beispiel einer Grafik .....	100
11.3.3 Das Prinzip der Verschlüsselung .....	101
11.3.4 Das Prinzip der Signatur .....	103
11.3.5 Lösung: Übergabe des symmetrischen Schlüssels .....	105
11.3.6 Erklärung der asymmetrischen Verschlüsselung mit trivialer Mathematik .....	106
11.4 Besonders drei Begriffe spielen eine außergewöhnliche Rolle .....	107
11.4.1 Authentizität .....	107
11.4.2 Integrität .....	107
11.4.3 Vertraulichkeit .....	107
11.5 Zusammenfassung Verschlüsselung .....	108
11.5.1 Erste Herausforderung .....	109
11.5.2 Zweite Herausforderung .....	109
11.6 Steganographie .....	110
12 Intranet und Extranet im VPN (Virtuelles Privates Netzwerk) .....	111
12.1 Die Begriffe Intranet und Extranet aus dem Blickwinkel von LAN 1 .....	111
12.2 Transportmodus vs. Tunnelmodus .....	112
12.2.1 Transportmodus (als Metapher) .....	112
12.2.2 Tunnelmodus (als Metapher) .....	112
13 Cloud .....	113
13.1 Sicherheit .....	114
13.2 Aufwand für die Nutzung einer Cloud .....	114
13.2.1 Linker Keil ("Anbieter") .....	114
13.2.2 Rechter Keil ("Kunde") .....	114

13.3 Beispiele.....	115
13.3.1 SaaS .....	115
13.3.2 PaaS .....	115
13.3.3 IaaS.....	115
13.4 Auflistung der Cloud-Typen .....	116
13.4.1 Public Cloud .....	116
13.4.2 Hybride Cloud.....	116
13.4.3 Community Cloud.....	116
13.4.4 Private Cloud.....	116
14 Phishing .....	117
14.1 Erklärung zu 14_Phishing_DHCP_Server_Teams.flis .....	117
15 ARP-Spoofing.....	118
15.1 Erklärung zu 15_ARP-Spoofing.flis .....	118
15.2 Schutz gegen ARP-Spoofing .....	118
16 E-Mailing .....	119
16.1 Erklärung zu 16_Alice_Mail_Bob_intern.flis.....	119
16.2 Alternative zu POP3 .....	119
17 Anhang.....	120
Ein Blick über den Tellerrand .....	120
Mit Papier und Schere .....	122

## 0 Quellen

IT-Handbuch Westermann 13.Auflage

### 0.1 Quellen aus dem Internet

- [de.wikipedia.org](https://de.wikipedia.org)
- [wut.de/download/print/e-58www-11-prde-000.pdf](https://wut.de/download/print/e-58www-11-prde-000.pdf)
- [wut.de/download/print/e-58www-20-prde-000.pdf](https://wut.de/download/print/e-58www-20-prde-000.pdf)
- [thomas-krenn.com/de/wiki/Kategorie:Netzwerk+Zubehör](https://thomas-krenn.com/de/wiki/Kategorie:Netzwerk+Zubeh%C3%B6r)
- [thomas-krenn.com/de/wiki/Kostenlose IT Bücher](https://thomas-krenn.com/de/wiki/Kostenlose_IT_B%C3%BCher)
- [youtube.com](https://youtube.com)
- [avm.de/ratgeber/filter=technologie](https://avm.de/ratgeber/filter=technologie)

### 0.2 Lernsoftware-filius.de

[Filius - Herunterladen](#)

Im weiteren Unterricht wird die „BFW-Edition“ von Filius gebraucht:

<https://git.michm.de/manne/filius-fork/-/releases>

#### 0.2.1 Simulationen von Filius

In den Simulationen von Filius gilt oft, aber nicht immer, folgende Zuordnung:

- die Netzwerke nutzen den IP-Adressraum: 192.168.x.y / 24
- das Standard-Gateway hat die IP-Adresse: 192.168.x.1 / 24
- (Achtung: Bei den Aufgaben der IHK wird oft die letzte Adresse des Netzsegmentes benutzt.)
- die Fritz!Box ist oft das Standard-Gateway
- der DHCP-Server hat die IP-Adresse: 192.168.x.11 / 24
- der DNS-Server hat die IP-Adresse: 192.168.x.12 / 24
- der Web-Server hat die IP-Adresse: 192.168.x.13 / 24

# 1 Allgemein

## 1.1 Geschichte

[de.wikipedia.org/wiki/Arpanet](https://de.wikipedia.org/wiki/Arpanet)

## 1.2 Leitungsvermittlung, Paketvermittlung

[netplanet.org/aufbau/netzwerk.shtml](https://netplanet.org/aufbau/netzwerk.shtml)

## 1.3 Kommunikationsprotokoll

[de.wikipedia.org/wiki/Kommunikationsprotokoll](https://de.wikipedia.org/wiki/Kommunikationsprotokoll)

- Das Zusammenspiel verschiedener Protokolle (ARP, IP, TCP, UDP, DHCP, DNS, Web), ist in der ersten Filius-Simulation zu beobachten.  
siehe /Filius\_Szenen/1.3\_PC\_DHCP\_DNS\_Web.flv
- Filius allgemein:
  - Start mit Klick auf den grünen Button
  - Rechts-Klick auf ein Netzwerkgerät => zeigt den Datenaustausch an

## 1.4 Netzwerk-Nodes

- Host: [de.wikipedia.org/wiki/Hostrechner](https://de.wikipedia.org/wiki/Hostrechner)
- Server: [de.wikipedia.org/wiki/Server](https://de.wikipedia.org/wiki/Server)

## 1.5 Vorteile von Netzwerken

- Schneller Datenaustausch ist möglich
- gemeinsame Ressourcennutzung ist möglich ( nur ein Drucker für alle Anwender im Büro)
- zentrale Datenspeicherung (Fileserver, NAS)
- Kostenersparnis (immer ein Argument), ....

## 1.6 Nachteile von Netzwerken

- schnelle Verbreitung von Schadsoftware ist möglich
- Spionage von innen und von außen ist möglich
- Kosten für Aufbau, Wartung und eventuell Personalkosten für die Administration, .....

## 1.7 Dimensionen von Netzwerken

- LAN: [de.wikipedia.org/wiki/Local Area Network](https://de.wikipedia.org/wiki/Local_Area_Network)
- MAN: [de.wikipedia.org/wiki/Metropolitan Area Network](https://de.wikipedia.org/wiki/Metropolitan_Area_Network)
- WAN: [de.wikipedia.org/wiki/Wide Area Network](https://de.wikipedia.org/wiki/Wide_Area_Network)  
[submarinecablemap.com](https://submarinecablemap.com)  
[he.net/3d-map](https://he.net/3d-map)  
[bbmaps.itu.int/bbmaps](https://bbmaps.itu.int/bbmaps)  
[map.kmcd.dev/?year=2025](https://map.kmcd.dev/?year=2025)

## 1.8 Topologien (allgemein)

[de.wikipedia.org/wiki/Topologie\\_\(Rechnernetz\)](https://de.wikipedia.org/wiki/Topologie_(Rechnernetz))

- Ringtopologie:  
[de.wikipedia.org/wiki/Topologie\\_\(Rechnernetz\)#Ring-Topologie](https://de.wikipedia.org/wiki/Topologie_(Rechnernetz)#Ring-Topologie)
- Bustopologie:  
[de.wikipedia.org/wiki/Topologie\\_\(Rechnernetz\)#Bus-Topologie](https://de.wikipedia.org/wiki/Topologie_(Rechnernetz)#Bus-Topologie)
- Sterntopologie:  
[de.wikipedia.org/wiki/Topologie\\_\(Rechnernetz\)#Stern-Topologie](https://de.wikipedia.org/wiki/Topologie_(Rechnernetz)#Stern-Topologie)

## 1.9 Echtzeitfähigkeit

[de.wikipedia.org/wiki/Echtzeitsystem#Harte, weiche und feste Echtzeit](https://de.wikipedia.org/wiki/Echtzeitsystem#Harte,_weiche_und_feste_Echtzeit)

## 1.10 Richtungsunabhängigkeit

[de.wikipedia.org/wiki/Duplex\\_\(Nachrichtentechnik\)](https://de.wikipedia.org/wiki/Duplex_(Nachrichtentechnik))

- Simplex: nur von A nach B (Einbahnstraße)
- Halbduplex: erst von A nach B, danach von B nach A (Baustellenampel), wird eingesetzt in der Bus- bzw. der Sterntopologie mit einem HUB
- Vollduplex: von A nach B und zeitgleich von B nach A (zweispurige Straße, Autobahn), wird eingesetzt in der Sterntopologie mit einem Switch

## 2 OSI-Schichtenmodell

### 2.1 Warum Schichten

Die Austauschbarkeit von Schichten ist möglich (z. B. IPv4 wird zukünftig durch IPv6 ersetzt).

### 2.2 verschiedene Schichtenmodelle

siehe Westermann Seite 590

- Nummerieren Sie bitte in der Spalte "ISO-OSI" (ganz links) die Schichten von "Bitübertragungsschicht" (blau) mit 1 bis "Anwendungsschicht" (gelb) mit 7
- In der Spalte "TCP/IP-Protokollstruktur" kennzeichnen Sie bitte die schwach erkennbare weiße Linie zwischen Schicht 1 und 2
- Schreiben Sie in Schicht 2 zusätzlich: MAC-Adresse
- Schreiben Sie in Schicht 1 zusätzlich: Netzwerkkarte

### 2.3 ISO-OSI-7-Schichten-Modell

siehe Westermann Seite 590

- Merksatz (von oben nach unten gesehen): => **A**lle **D**eutschen **S**chüler **T**rinken **V**erschiedene **S**orten **B**rause (oder andere Getränke, die mit „B“ beginnen).

### 2.4 TCP/IP-Modell

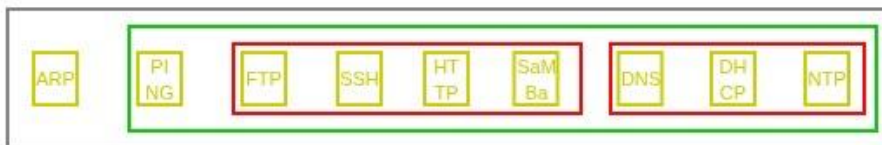
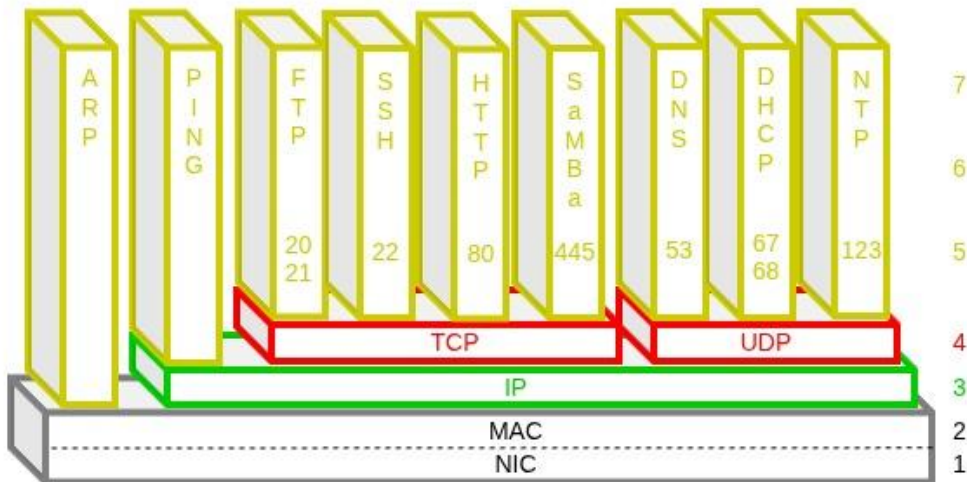
siehe Westermann Seite 590 Spalte "TCP/IP-Protokollstruktur"

[de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell](https://de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell)

==>> das am häufigsten eingesetzte Modell

### 2.5 TCP/IP-Modell und Schokolade

- Ganz innen die Schokolade => die Daten (Schichten 5..7 des TCP/IP-Modells)
- Drum herum das Silberpapier => TCP oder UDP (Schicht 4 des TCP/IP-Modells)
- Außen herum das bunte Papier => IP (Schicht 3 des TCP/IP-Modells)
- Die gesamte Tafel Schokolade stecken wir in eine Tüte => MAC-Adresse (Schicht 2 des TCP/IP-Modells)
- Der Einkaufswagen transportiert die Tüte mit der Schokolade (Schicht 1 des TCP/IP-Modells)
- Der Weg, auf dem wir unterwegs sind (Schicht "0" des TCP/IP-Modells)



### 3 Schicht Null

siehe Westermann Seite 590 ganz unten in weißer Farbe

#### 3.1 Koaxialkabel

[de.wikipedia.org/wiki/Koaxialkabel](https://de.wikipedia.org/wiki/Koaxialkabel)

- veraltete Technologie
- wurde für Bustopologie verwendet
- wurde als 10Base2 bezeichnet (10 Mbit/s Übertragungsgeschwindigkeit, Basisband, 200 yard lang)

#### 3.2 Twisted-Pair-Kabel

siehe Westermann Seiten 234 und 421

[de.wikipedia.org/wiki/Twisted-Pair-Kabel](https://de.wikipedia.org/wiki/Twisted-Pair-Kabel)

[de.wikipedia.org/wiki/Ethernet#Formate der Ethernet-Datenübertragungsblöcke und das Typfeld](https://de.wikipedia.org/wiki/Ethernet#Formate_der_Ethernet-Datenübertragungsblöcke_und_das_Typfeld)

##### 3.2.1 CAT 3

- für 10BASE-T (10 Mbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1,2,3,6 werden genutzt

##### 3.2.2 CAT 5

- für 100BASE-Tx (100 Mbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1,2,3,6 werden genutzt

##### 3.2.3 CAT 5e

- für 1000BASE-T (1000 Mbit/s = 1 Gbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1 bis 8 werden genutzt
- wird selten in Deutschland eingesetzt

##### 3.2.4 CAT 6

- für 1000BASE-T (1000 Mbit/s = 1 Gbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1 bis 8 werden genutzt
- für NBASE-T (2500 Mbit/s = 2,5 Gbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1 bis 8 werden genutzt
- für NBASE-T (5000 Mbit/s = 5 Gbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1 bis 8 werden genutzt

### 3.2.5 CAT 6<sub>A</sub>

- für 10GBASE-T (10000 Mbit/s = 10 Gbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1 bis 8 werden genutzt

### 3.2.6 CAT 7 / 8

- für 25GBASE-T (25000 Mbit/s = 25 Gbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1 bis 8 werden genutzt
- für 40GBASE-T (40000 Mbit/s = 40 Gbit/s Übertragungsgeschwindigkeit, Basisband, Twisted Pair), die Adern 1 bis 8 werden genutzt

## 3.3 Lichtwellenleiter (LWL)

siehe Westermann Seite 190

[science.lu/de/wasser-experiment/leite-licht-mit-wasser-um-die-ecke](http://science.lu/de/wasser-experiment/leite-licht-mit-wasser-um-die-ecke)  
[youtube.com/watch?v=0MwMkBet\\_5I](https://youtube.com/watch?v=0MwMkBet_5I)

### 3.3.1 Multimode

- "Mehrmoden-Stufenfaser" (obere Abbildung): veraltet, zeigt aber das Prinzip sehr gut
- "Mehrmoden-Gradientenfaser" (mittlere Abbildung), aktuelle Technik mit Kerndurchmesser = 50 µm und einem Manteldurchmesser = 125 µm

### 3.3.2 Singlemode

Einmoden-Stufenfaser" (untere Abbildung): aktuelle Technik mit Kerndurchmesser = 10 µm und Manteldurchmesser = 125 µm, wird auch manchmal "Monomode" genannt

## 3.4 Dämpfung (speziell LWL)

[de.wikipedia.org/wiki/Dämpfung](http://de.wikipedia.org/wiki/D%C3%A4mpfung)

siehe Westermann Seite 189

- Dämpfung ist das Gegenteil von Verstärkung
- Dämpfung ändert sich bei verschiedenen Wellenlängen, könnten wir dieses Licht sehen, würden wir von verschiedenen Farben sprechen

### 3.4.1 Welches Licht können wir sehen?

siehe Westermann Seite 204:

==>> NICHT! in eine Glasfaser schauen: unsichtbares Laserlicht!

### **3.4.2 Zuordnung in der Praxis**

In der Praxis gibt es folgende Zuordnung (Seite 189)

- Bereich bei 850 nm wird als "SX" bezeichnet
- Bereich bei 1300 nm wird als "LX" bezeichnet
- Bereich bei 1550 nm wird als "ZX" bezeichnet

### **3.4.3 Geschwindigkeiten und Entfernungen von LWL**

siehe Westermann Seite 422 (unten)

## **3.5 Strukturierte Verkabelung**

siehe Westermann Seite 421

[de.wikipedia.org/wiki/Strukturierte\\_Verkabelung](https://de.wikipedia.org/wiki/Strukturierte_Verkabelung)

(siehe Bild auf dieser Webseite)

### **3.5.1 Bindeglied zwischen TP-Kabel und LWL**

Medienkonverter: [de.wikipedia.org/wiki/Medienkonverter](https://de.wikipedia.org/wiki/Medienkonverter)

SFP (Mini-GBIC): [de.wikipedia.org/wiki/Small\\_Form-factor\\_Pluggable](https://de.wikipedia.org/wiki/Small_Form-factor_Pluggable)

### 3.6 Einfache Regel für die Verkabelung

- Bis 100 m nutzt man Twisted-Pair-Kabel.
- Bis 2000 m (praktisch bis 550 m) nutzt man Multimode-LWL.
- Darüber nutzt man Singlemode-LWL (Monomode)

### 3.7 Wireless LAN (WLAN)

[de.wikipedia.org/wiki/Wireless Local Area Network#Standards](https://de.wikipedia.org/wiki/Wireless_Local_Area_Network#Standards) nach IEEE 802.11

[heise.de/select/ct/2019/2/1546773697424925](https://heise.de/select/ct/2019/2/1546773697424925)

c't iperf für Windows

siehe /Filius\_Szenen/3.7\_WLAN.flv

#### 3.7.1 Normen von WLAN und deren (Vermarktungs-Name)

- IEEE 802.11 a => 5 GHz
- IEEE 802.11 b => 2,4 GHz
- IEEE 802.11 g => 2,4 GHz
- (WIFI 4) IEEE 802.11 n => 2,4 und 5 GHz
- (WIFI 5) IEEE 802.11 ac => 5 GHz
- (WIFI 6) IEEE 802.11 ax => 2,4 und 5 GHz
- (WIFI 6E) IEEE 802.11 ax => 2,4 und 5 GHz und 6 GHz
- (WIFI 7) IEEE 802.11 be => 2,4 und 5 GHz und 6 GHz (die Frequenzbänder können zusammengefasst werden)

[Mesh-WLAN – Wikipedia](#)

#### 3.7.2 MIMO / MU-MIMO

siehe Westermann Seite 218 (Abbildung unten rechts)

[de.wikipedia.org/wiki/MIMO](https://de.wikipedia.org/wiki/MIMO) (Nachrichtentechnik)

Begriffserklärung allgemein "Multiple In Multiple Out":

Über mehrere Antennen werden möglichst gleichzeitig mehrere parallele Datenströme geleitet.

=> Als würde man gleichzeitig durch mehrere Strohhalm trinken.

### 3.7.2.1 MIMO

- Der AP (z. B. Fritz!Box) besitzt angenommen 4 Antennen und soll mit Laptop A (mit je 2 Antennen) und mit Laptop B (auch mit je 2 Antennen) kommunizieren. Der AP nutzt nur 2 seiner 4 Antennen.
- Erst kommuniziert er mit Laptop A mit dessen 2 Antennen, dann unterbricht der AP diese Verbindung.
- Anschließend bedient der AP für eine gewisse Zeit Laptop B mit dessen 2 Antennen.
- Dann wird auch diese Verbindung nach einer gewissen Zeit wieder unterbrochen und der AP bedient erneut Laptop A.

### 3.7.2.2 MU-MIMO

- Der AP (z. B. Fritz!Box) besitzt wieder angenommen 4 Antennen und soll mit Laptop A (mit je 2 Antennen) und mit Laptop B (auch mit 2 Antennen) kommunizieren (siehe oben).
- Der AP nutzt alle 4 Antennen und kommuniziert zeitgleich mit Laptop A mit dessen 2 Antennen und mit Laptop B mit dessen 2 Antennen.

### 3.7.3 Sicherheit im WLAN (kurz und knapp)

- Das WLAN sollte / muss verschlüsselt werden! (mindestens WPA 2, WPA 3 empfohlen)
- Es darf heute nur noch ein Sicherheitskonzept eingesetzt werden, das zum Zeitpunkt des Kaufes der WLAN-Komponente marktüblich war!
- Es ergibt absolut Sinn, das WLAN mit verschiedenen SSIDs zu betreiben (Home, Gäste).

siehe /Filius\_Szenen/3.7.3\_NAT\_extern\_DNS\_SSID\_1\_2.flv

## 3.8 Dezibel [dB]

Ist ein Faktor (wie viel mal mehr oder weniger), bei Leistung gilt:

10 dB => Faktor	10
13 dB => Faktor	20
16 dB => Faktor	40
19 dB => Faktor	80
20 dB => Faktor	100
30 dB => Faktor	1000
40 dB => Faktor	10000

### 3.9 Antennengewinn

Eine Antenne kann nichts "gewinnen", sie konzentriert nur die abgestrahlte Energie.

#### 3.9.1 Beispiel: Taschenlampe

Wer sie noch kennt, die klassische Metall-Stabtaschenlampe mit einer "Glühbirne":

- ohne Reflektor => überall ist es "nicht wirklich hell", niemand wird geblendet
- mit Reflektor => in einem kleinen Bereich ist es sehr hell und kann stark blenden
- Nach diesem Prinzip arbeitet auch eine Antenne, sie ist sozusagen eine Art "Reflektor".

#### 3.9.2 EIRP, ERP „das Blenden“

- Um das "Blenden" (zu viel abgestrahlte Energie auf einen zu kleinen Punkt) zu verhindern, gilt:
- EIRP, sozusagen die Leistung ohne Reflektor:
- ERP, sozusagen die noch erlaubte Leistung mit Reflektor:

Berechnung an einem konkreten Beispiel:

gegeben: Antennengewinn = 13 dB, EIRP = 100 mW (vom Gesetzgeber festgelegt)

gesucht: ERP

Lösung:

13 dB => Faktor 20

ERP = 100 mW / 20

ERP = 5 mW

Der Sender (Fritzbox, AP) darf nur 5 mW Leistung an die Antenne „schicken“.

Eine höhere Leistung wäre illegal!

## 4 Ethernet-Frame

[wut.de/download/print/e-58www-11-prde-000.pdf](http://wut.de/download/print/e-58www-11-prde-000.pdf) Seite 31 bis 34

siehe Westermann Seite 579 "Rahmenformate" und Seite 581 "Rahmenstruktur"

### 4.1 Präambel

[de.wikipedia.org/wiki/Synchronisation](http://de.wikipedia.org/wiki/Synchronisation)

### 4.2 "Rahmenformate Ethernet II"

siehe Westermann Seite 579, 581

- ist der bevorzugte Rahmentyp
- Im Feld "DATA" befinden sich IP, TCP/UDP und die eigentlichen Daten

### 4.3 CRC/FCS

[de.wikipedia.org/wiki/Blockprüfzeichenfolge](http://de.wikipedia.org/wiki/Blockprüfzeichenfolge)

siehe Westermann Seite 580 "FCS"

### 4.4 Problem der Größe des Ethernet-Frames

[de.wikipedia.org/wiki/Maximum Transmission Unit](http://de.wikipedia.org/wiki/Maximum_Transmission_Unit)

- Der meist genutzte Frame kann nur 1500 Byte an Daten aufnehmen, aber darin stecken auch noch die Header von IP und TCP/UDP und möglicherweise HTTP  
siehe Westermann Seite 581 "Rahmenstruktur"  
=> Somit bleiben nur ca. 1460 Byte pro Frame (Datenpaket) für die reinen Daten übrig.

siehe [de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell](http://de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell)

siehe Grafik unten auf der Webseite:

"Aufbau eines Ethernet-Frames mit maximalen IPv4- / TCP-Daten"

=> Für heutige Anwendungen (Dateigrößen) ist der Ethernet-Frame eigentlich zu klein.

#### 4.4.1 Jumbo-Frame

Jumbo-Frame: [de.wikipedia.org/wiki/Jumbo Frames](http://de.wikipedia.org/wiki/Jumbo_Frames)

- Alle Netzwerkkomponenten müssen durchgehend den Jumbo-Frame unterstützen.
- setzt sich seit Jahren in der Praxis nicht durch

## 5 Sniffer

### 5.1 Rechtliches

Wenn ein Sniffer eingesetzt werden soll, gilt Folgendes zu beachten:

- Einsatz in der schulischen Ausbildung problemlos möglich, wenn keine Daten ausspioniert werden.
- Einsatz im Unternehmen nur möglich, wenn Vorgesetzter und (wenn vorhanden) Betriebsrat informiert werden
- ==>> niemals illegal einsetzen, sonst „weht ein eisiger Wind“ durch Ihren Arbeitsvertrag

### 5.2 Einfacher Sniffer

Einfacher Sniffer:

- Packetyzer (alt aber gut) [sourceforge.net/projects/packetyzer/files/](https://sourceforge.net/projects/packetyzer/files/)
- findet seinen Einsatz bei uns im Unterricht

### 5.3 Wireshark

Wer mehr möchte und sich später intensiv mit dem Thema auseinandersetzen möchte:

Wireshark

[wireshark.org/#download](https://wireshark.org/#download)

### 5.4 Netzwerkkarte “Freizügiger Modus”

- Treiber, um die Netzwerkkarte unter Windows in den "freizügigen Modus" zu versetzen:  
[npcap.com/#download](https://npcap.com/#download)
- ==>> unter Windows 11 NICHT! winpcap benutzen  
[de.wikipedia.org/wiki/Promiskuitiver Modus](https://de.wikipedia.org/wiki/Promiskuitiver_Modus)

### 5.5 Zu empfehlende Konsolen-Sniffer:

#### 5.5.1 tcpdump (Linux)

- Die optimale Wahl, wenn man skripten möchte.
- In Verbindung mit dem Tool „geoiplookup“ kann man erkennen, wohin der Rechner gerade Daten versendet.

#### 5.5.2 Windump

Der Nachbau von „tcpdump“ für User, die in der Netzwerkadministration noch immer Windows einsetzen (müssen).

## 6 Schicht Eins

[de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell](https://de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell)  
[de.wikipedia.org/wiki/Netzwerkkarte](https://de.wikipedia.org/wiki/Netzwerkkarte)

### 6.1 Netzwerkkarte

Koppelement zwischen der Rechnerhardware und dem Netzwerk

#### 6.1.1 Hauptaufgaben der Netzwerkkarte

Pegelanpassung

- Medienkonvertierung (elektrisch  $\Leftrightarrow$  Licht, elektrisch  $\Leftrightarrow$  Funk)
- Erzeugung und Kontrolle der Prüfsumme (CRC) des Frames
- Steuerung des Zugriffs auf das Medium

### 6.2 Netzzugriffsverfahren

#### 6.2.1 Kollisionserkennung (CSMA/CD)

[de.wikipedia.org/wiki/Carrier Sense Multiple Access/Collision Detection](https://de.wikipedia.org/wiki/Carrier_Sense_Multiple_Access/Collision_Detection)

=> „Lass die Frames kollidieren, Hauptsache es merkt jemand und reagiert darauf“.

#### 6.2.2 Kollisionsvermeidung (CSMA/CA)

[de.wikipedia.org/wiki/Carrier Sense Multiple Access/Collision Avoidance](https://de.wikipedia.org/wiki/Carrier_Sense_Multiple_Access/Collision_Avoidance)

=> „Vermeide, dass es zu einer Kollision der Frames kommt.“

Merksatz:

CSMA/CD => D für „Drahtgebundene“ Netze

CSMA/CA => A für Netze mit „Antennen“

### **6.3 HUB:**

[de.wikipedia.org/wiki/Hub](https://de.wikipedia.org/wiki/Hub) (Netzwerktechnik)

- Vorgänger eines Switches
- verteilt alle Datenpakete an alle Netzwerkgeräte => sehr viel unnötiger Datenverkehr im Netzwerk
- Sniffing sehr gut möglich
- nur bis 100 Mbit/s einsetzbar, da Hubs nicht für 1 Gbit/s hergestellt wurden
- nur Halbduplex ist möglich

## 7 Schicht Zwei

[de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell](https://de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell)

### 7.1 MAC-Adresse

[de.wikipedia.org/wiki/MAC-Adresse](https://de.wikipedia.org/wiki/MAC-Adresse)

- sollte weltweit einmalig sein
- wird hexadezimal dargestellt
- ist 48 bit lang
- Die ersten 24 bit definieren den Hersteller.
- Die zweiten 24 bit sind eine Zählnummer des jeweiligen Herstellers.
- Das 7. bit zeigt an, ob es sich um eine physische oder eine virtuelle Netzwerkkarte handelt.

### 7.2 Switch

- Verteilt anhand der MAC-Adresse das Datenpaket (Frame) nur an den Host, für den das Datenpaket bestimmt ist.
- Ausnahme: MAC-Adresse FF:FF:FF:FF:FF:FF => Ruf an alle Netzwerkkarten (Schicht 2 Broadcast).
- Unnötiger Datenverkehr im Netzwerk wird vermieden.
- Ist heute die übliche Komponente im Netzwerk.

siehe /Filius\_Szenen/7.2\_PCs\_2\_Switches.flv

siehe /Filius\_Szenen/7.2\_Switch\_4\_PCs.flv

=> Links-Klick auf den jeweiligen Switch, zeigt die „SAT-Tabelle“ des Switches an.

#### 7.2.1 Wie wird die SAT-Tabelle aufgebaut?

- ARP-REQUEST: „Wer hat die IP-Adresse a.b.c.d? Ich brauche deine MAC-Adresse.“
- ARP-REPLY: „Ich habe die gesuchte IP-Adresse a.b.c.d!. Meine MAC-Adresse lautet .....!“

#### 7.2.2 Unmanaged Switch (Desktop Switch)

- Zugriff auf den Switch nicht möglich und auch nicht nötig
- Sniffing nicht ganz so einfach möglich
- oft haben unmanaged Switches nur bis zu 24 Ports

### **7.2.3 Managed Switch**

Zugriff auf den Switch meist per Web-Frontend oder Konsole.

- Umfangreiche Einstellungen für jeden einzelnen Port möglich.

### **7.2.4 Features moderner (managed) Switches:**

#### **7.2.4.1 Port-Mirroring**

- Spiegelung des Datenverkehrs eines Ports an einen zweiten Port für Analyse (Sniffing)

#### **7.2.4.2 Link Aggregation**

[de.wikipedia.org/wiki/Link\\_Aggregation](https://de.wikipedia.org/wiki/Link_Aggregation)

- Mehrere physische Ports werden zu einem logischen Port zusammengefasst, um den Datendurchsatz zu erhöhen (erinnert an MIMO).
- Wird im Linux-Umfeld auch Bonding genannt.
- Wird allgemein oft als Trunking bezeichnet. => Vorsicht: Cisco versteht darunter etwas anderes.
- „Pseudo“-Link-Aggregation => Aufteilung des Datenstroms anhand der MAC-Adresse

### 7.2.4.3 Power over Ethernet

siehe Westermann Seite 585

[de.wikipedia.org/wiki/Power over Ethernet](https://de.wikipedia.org/wiki/Power_over_Ethernet)

==>> Die Leistung der (PSE)-Versorgung muss zur Berechnung genutzt werden!

==>> Die Gesamtleistung des Switches bestimmt, wie viele Ports für PoE genutzt werden können.

PoE – Normen

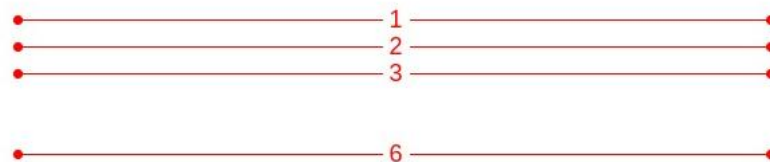
Tabelle "Vergleich der PoE-Standards"

- 802.3af
- 802.3at
- 802.3bt

„passive-PoE-Adapter“ => nur 10/100 Mbit/s möglich! siehe folgendes Bild:

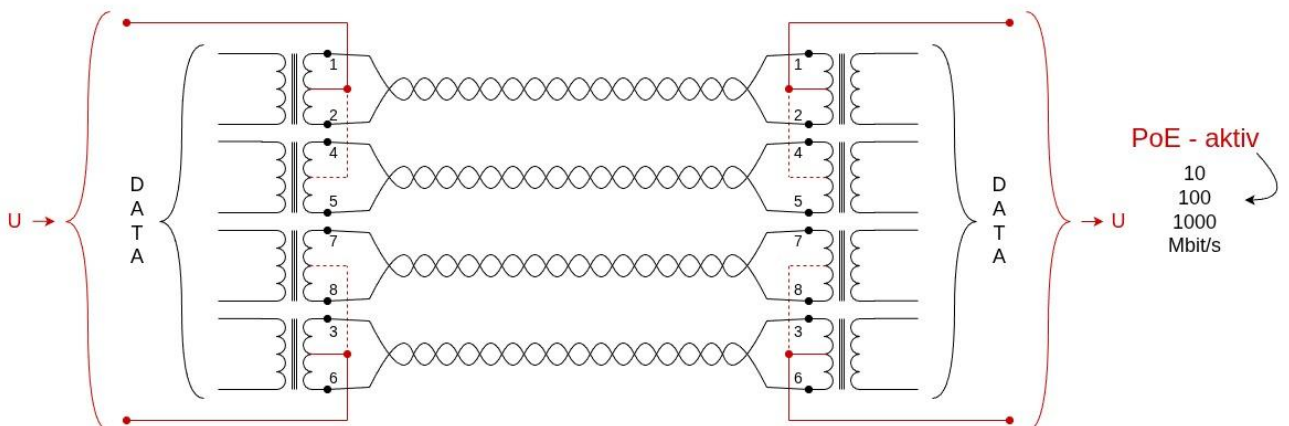
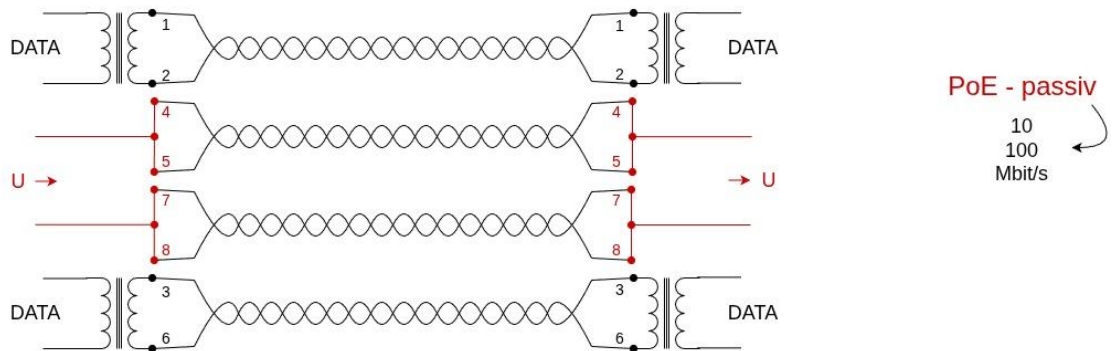
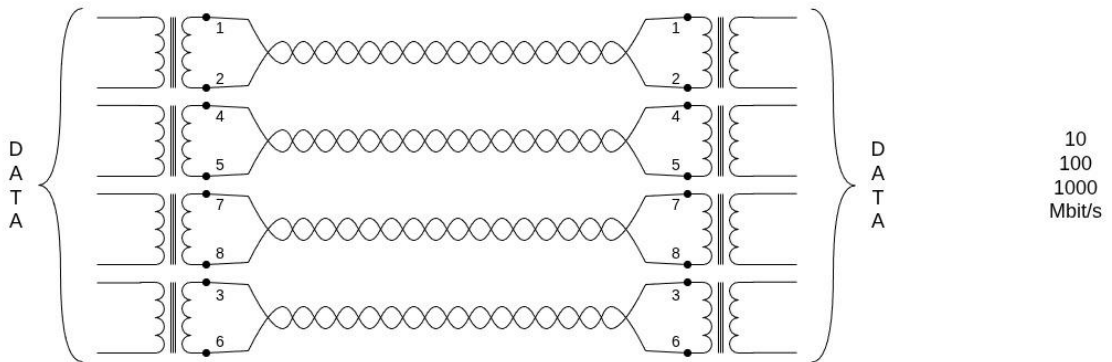
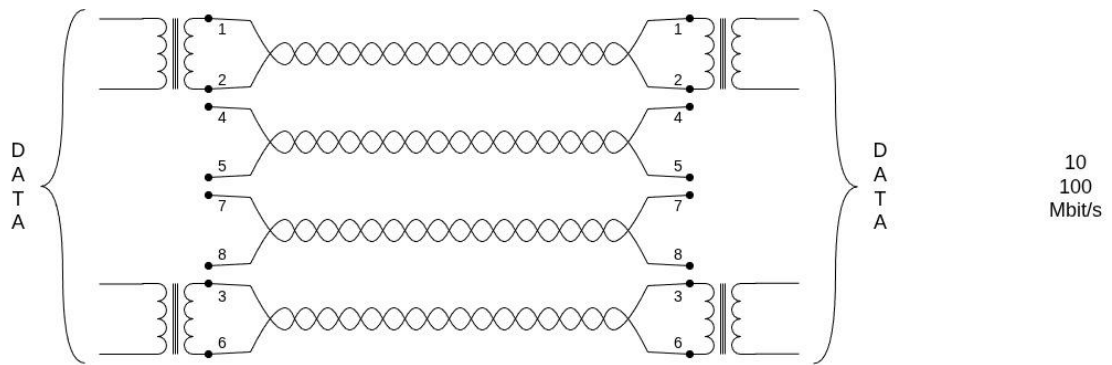


$$1 + 2 + 3 + 6 + 4 + 5 + 7 + 8 \rightarrow 10 / 100 / \underline{1000} \text{ Mbit/s}$$

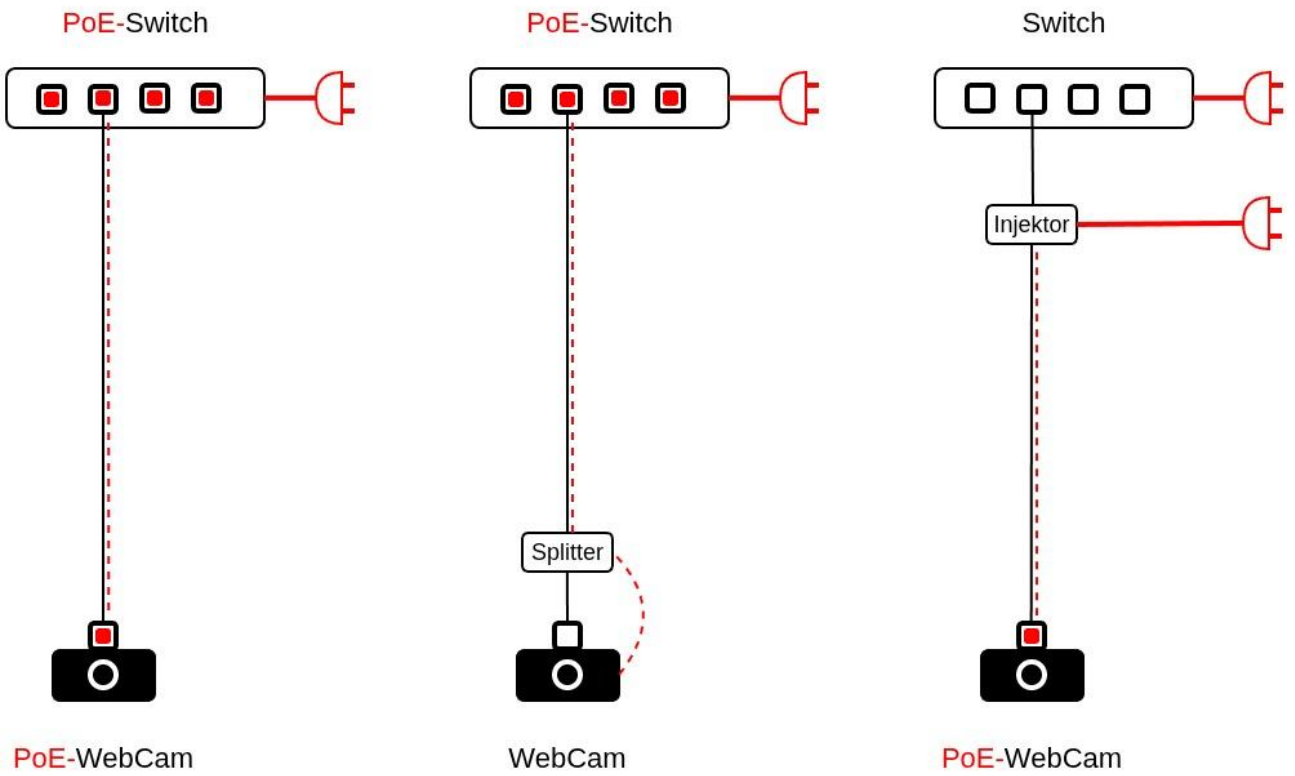


$$1 + 2 + 3 + 6 \rightarrow 10 / \underline{100} \text{ Mbit/s}$$





□ Daten      ■ PoE      □ Daten + PoE



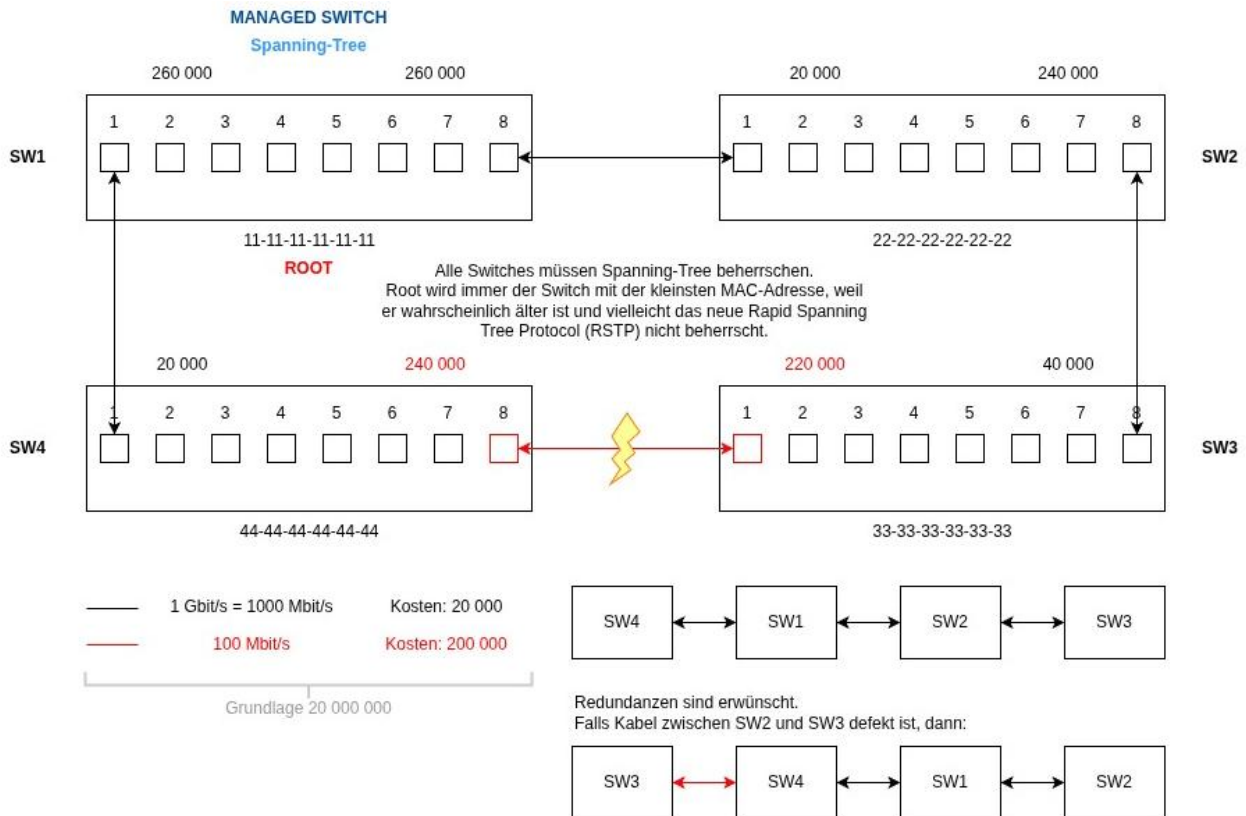
## 7.2.4.4 Spanning Tree

siehe:

[de.wikipedia.org/wiki/Spanning\\_Tree\\_Protocol](https://de.wikipedia.org/wiki/Spanning_Tree_Protocol)

[de.wikipedia.org/wiki/Redundanz\\_\(Technik\)](https://de.wikipedia.org/wiki/Redundanz_(Technik))

siehe /Filius\_Szenen/7.2.4.3\_Spanning\_Tree\_4\_PCs.flv



- zur Vermeidung von Schleifen unter den Switches
- zum Aufbau von redundanten Wegen zwischen den Switches
- sogenannte Kosten als Grundlage der Berechnung
- Die langsamste Verbindung wird unterbrochen => hier zwischen SW3 und SW4.
- Falls die Verbindung zwischen SW2 und SW3 ausfällt => Neuberechnung:  
=> Die langsame Verbindung zwischen SW3 und SW4 wird wieder in Betrieb genommen

Spanning Tree - Kosten,

Erklärungsversuch: Wartezeit (in Millisekunden) für die Übertragung einer Datei

- 2,5 GByte (nicht GiByte!) == 20 Gbit über 1 Gbit/s => 20 Sekunden == 20000 Millisekunden
- 2,5 GByte (nicht GiByte!) == 20 Gbit über 100 Mbit/s => 200 Sekunden == 200000 Millisekunden

==>> vom Root aus wird gerechnet

#### 7.2.4.5 VLAN

Westermann Seite 597 "VLAN"

[de.wikipedia.org/wiki/Virtual\\_Local\\_Area\\_Network](https://de.wikipedia.org/wiki/Virtual_Local_Area_Network)

[heise.de/ct/artikel/VLAN-Virtuelles-LAN-221621](https://heise.de/ct/artikel/VLAN-Virtuelles-LAN-221621)

[thomas-krenn.com/de/wiki/VLAN Grundlagen](https://thomas-krenn.com/de/wiki/VLAN_Grundlagen)

- Ein physisches LAN wird in mehrere logische LANs aufgeteilt.
- eine sehr moderne Form, um Zugriffe im LAN steuern zu können

VLAN-Tagging:

Die Verbindung zwischen VLAN-Switches erfolgt üblicherweise durch "Tagged VLANs" über nur einen Port.

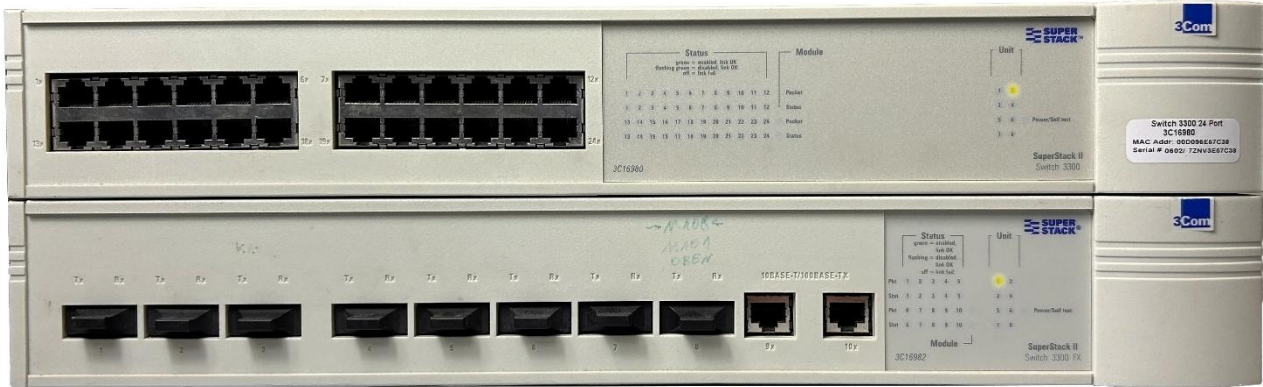
=> "Flaschenhals" => Link Aggregation einsetzen!

Durch das Taggen wird der Ethernet-Frame um 4 Byte erweitert:

- Der Frame kann von 1518 Byte auf 1522 Byte anwachsen.
- Alle Übertragungsgeräte (z. B. Medienkonverter) zwischen den Switches müssen "VLAN-transparent" sein.

### 7.2.4.6 Stackable Switch

- Zwei oder mehrere stapelbare (physische) Switches werden zu einem logischen Switch zusammengefasst.
- Dafür werden proprietäre (sehr teure) Kabel benötigt.



- Beide Switches verhalten sich dann wie ein einziger Switch, auch bei der Konfiguration.
- Es gibt keinen Flaschenhals bei der Datenübertragung zwischen den Switches.
- grünes Kabel vom oberen Switch (Slave) => Verteiler im unteren Switch
- rotes Kabel aus dem Verteiler => Anschluss im unteren Switch (Master)

## 8 Schicht Drei

[de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell](https://de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell)

Themen aus Schicht 3:

- IPv4 => Adressen, besondere Adressen, Subnetting
- IPv6 => Adressen, besondere Adressen, Subnetting
- Vergleich der Header von IPv4 und IPv6
- DHCP
- Namensauflösung
- Routing

Übungen zum Subnetting:

[subnetipv4.com](http://subnetipv4.com)

[www.easy-network.de/subnetting-aufgaben.html](http://www.easy-network.de/subnetting-aufgaben.html)

[xinix.net/index.php/Subnetting](http://xinix.net/index.php/Subnetting) Berechnen Aufgaben

### 8.1 IPv4

#### 8.1.1 IP-Adresse

- Die IP-Adresse ist 32 bit lang.
- Die 32 bit der Adresse werden in 4 Oktette zu je 8 bit aufgeteilt.
- Die Oktette werden durch einen Punkt voneinander getrennt.
- Die 8 bit je Oktett werden dezimal dargestellt.
- Wir sehen:  
192 . 168 . 1 . 3
- Als Kompromiss zwischen der Speicherung der Daten im Computer und uns schreiben wir auch nach jedem binären Oktett einen Punkt:  
11000000.10101000.00000001.00000011
- Im Speicher des Computers steht nur ein „Binär-Wurm“, der uns Menschen verwirrt:  
110000001010100000000000100000011

### 8.1.2 Subnetzmaske

- Die Subnetzmaske ist auch 32 bit lang.
- Es werden zwei Schreibweisen der Subnetzmaske angewendet:  
 Klassische (dezimale) Schreibweise:
  - Die Subnetzmaske wird auch in 4 Oktette zu je 8 bit aufgeteilt.
  - Die Oktette werden auch durch einen Punkt voneinander getrennt.
  - Die 8 bit je Oktett werden auch dezimal dargestellt.
 CIDR-Schreibweise (modern):
  - Die Anzahl der binären "Einsen" wird hinter einem "/" geschrieben.

### 8.1.3 IPv4 Klassen (Class)

[de.wikipedia.org/wiki/Classless Inter-Domain Routing](https://de.wikipedia.org/wiki/Classless_Inter-Domain_Routing)

Erklärungsversuch der IPv4-Klassen im Vergleich mit alter drahtgebundener Telefonie:



### 8.1.3.1 Class A

- Das Oktett 1 liegt zwischen 0 und 127 (128 verschiedene Zahlen).
- Die Oktette 2, 3, 4 liegen zwischen 0 und 255 (jeweils 256 verschiedene Zahlen pro Oktett)
- Die Standardsubnetzmaske lautet:
- klassische (dezimale) Schreibweise: 255.0.0.0
- CIDR Schreibweise: /8
- Somit ergeben sich:  
128 Netze (0 – 127) aus dem Oktett 1 (zählt durch die Subnetzmaske zur Netz-ID).  
Jedes Netz hat 16.777.216 (-2) Adressen ( $256 * 256 * 256 = 2^{24}$ ) aus den Oktetten 2, 3, 4

### 8.1.3.2 Class B

- Das Oktett 1 liegt zwischen 128 und 191 (64 verschiedene Zahlen).
- Die Oktette 2, 3, 4 liegen zwischen 0 und 255 (jeweils 256 verschiedene Zahlen pro Oktett).
- Die Standardsubnetzmaske lautet:
- klassische (dezimale) Schreibweise: 255.255.0.0
- CIDR Schreibweise: /16
- Somit ergeben sich:  
64 Netze (128 – 191) aus dem Oktett 1 \* 256 Netze (0 – 255) aus dem Oktett 2  
= 16.384 Netze (Oktett 1 und 2 zählen durch die Subnetzmaske jetzt zur Netz-ID).  
Jedes Netz hat 65.536 (-2) Adressen ( $256 * 256 = 2^{16}$ ) aus Oktett 3 und 4

### 8.1.3.3 Class C

- Das Oktett 1 liegt zwischen 192 und 223 (32 verschiedene Zahlen).
- Die Oktette 2, 3, 4 liegen zwischen 0 und 255 (jeweils 256 verschiedene Zahlen pro Oktett).
- Die Standardsubnetzmaske lautet:
- klassische (dezimale) Schreibweise: 255.255.255.0
- CIDR-Schreibweise: /24
- Somit ergeben sich:  
32 Netze (192 – 223) aus dem Oktett 1 \* 256 Netze (0 – 255) aus dem Oktett 2  
\* 256 Netze (0 – 255) aus dem Oktett 3  
= 2.097.152 Netze (Oktett 1 und 2 und 3 zählen durch die Subnetzmaske zur Netz-ID).  
Jedes Netz hat 256 (-2) Adressen ( $256 = 2^8$ ) aus Oktett 4.

### 8.1.4 Besonderheiten

- In jedem Netz (auch in jedem Subnetz) können 2 Adressen NICHT benutzt werden.
- Der Anfang des Netzes (unterste Adresse) bezeichnet das Netz (eine Art "Joker").
- Das Ende des Netzes (höchste Adresse) ist der Broadcast ("Ruf an alle im Netz").
- Soll bei IPv4 und auch bei IPv6 ein ganzes Netz benannt werden, so wird der "Joker" benutzt:  
0 ist der Beginn des Netzes und bedeutet: 0 – 255

Beispiele aus Class A, B, C

- 10.0.0.0 /8 bedeutet: 10 . 0 – 255 . 0 – 255 . 0 – 255
- 172.17.0.0 /16 bedeutet: 172 . 17 . 0 – 255 . 0 – 255
- 192.168.1.0 /24 bedeutet: 192 . 168 . 1 . 0 – 255

==>> Im weiteren Verlauf wird häufig die CIDR-Schreibweise verwendet.

## 8.1.5 Besondere IPv4 Adressen (Auszug)

### 8.1.5.1 Class A

- 0.0.0.0 ist eine un spezifizierte Adresse ("Ich weiß nicht, wer ich bin.")

- 10.0.0.0 /8 ist ein privater Adressbereich in Class A

[de.wikipedia.org/wiki/Private IP-Adresse](https://de.wikipedia.org/wiki/Private_IP-Adresse)

- 127.0.0.1 localhost oder auch loopback genannt

[de.wikipedia.org/wiki/Localhost](https://de.wikipedia.org/wiki/Localhost)

Adresse der virtuellen Netzwerkkarte im netzwerkfähigen Rechner

=> Datenaustausch darüber ist innerhalb des Betriebssystems und von installierten Anwendungen möglich.

### 8.1.5.2 Class B

- 169.254.0.0 /16 ist eine "Link Local" (APIPA) - Adresse

[de.wikipedia.org/wiki/Private IP-Adresse](https://de.wikipedia.org/wiki/Private_IP-Adresse)

Sieht man häufig, wenn DHCP bei IPv4 nicht erfolgreich ist.

Das entsprechende Gerät (PC, Drucker) gibt sich SELBST eine Adresse aus diesem Bereich, siehe Zeroconf: [de.wikipedia.org/wiki/Zeroconf](https://de.wikipedia.org/wiki/Zeroconf)

- 172.16.0.0 bis 172.31.0.0 /16 ist ein privater Adressbereich in Class B

[de.wikipedia.org/wiki/Private IP-Adresse](https://de.wikipedia.org/wiki/Private_IP-Adresse)

Wird oft auch so geschrieben: 172.16.0.0 / 12 (rechnen Sie mal nach!)

### 8.1.5.3 Class C

- 192.168.0.0 /16 ist ein privater Adressbereich in Class C

[de.wikipedia.org/wiki/Private IP-Adresse](https://de.wikipedia.org/wiki/Private_IP-Adresse)

## 8.1.6 Standardsubnetzmaske vs. Subnetzmaske

Betrachten wir die privaten Adressbereiche aus Class A, B und C:

- 10.0.0.0 /8 => 1 Netz mit 16.777.216 (-2) Adressen
- 172.16.0.0 bis 172.31.0.0 /16 => 16 Netze mit je 65.536 (-2) Adressen
- 192.168.0.0 /24 => 256 Netze mit je 256 (-2) Adressen

### 8.1.6.1 Problem 1

- Gesucht wird eine Lösung für 50 private Netze mit je 300 Adressen:

10.0.0.0 /8 => funktioniert nicht, da es nur 1 Netz gibt

172.16.0.0 bis 172.31.0.0 /16 => funktioniert nicht, da es nur 16 Netze gibt

192.168.0.0 /24 => funktioniert nicht, da die Netze zu klein sind

- Lösung: 10.0.0.0 /16

Nutzung des privaten Netzes aus Class A

Nutzung der Standardsubnetzmaske aus Class B

Somit ergibt sich:

1 Netz (10) aus dem Oktett 1 \* 256 Netze (0 – 255) aus dem Oktett 2

= 256 Netze (Oktett 1 und 2 zählen durch die Subnetzmaske jetzt zur Netz-ID).

=> Jedes Netz hat 65.536 (-2) Adressen ( $256 * 256 = 2^{16}$ ) aus Oktett 3 und 4.

### 8.1.6.2 Problem 2

- Gesucht wird eine Lösung für 300 private Netze mit je 50 Adressen:

10.0.0.0 /8 => funktioniert nicht, da es nur 1 Netz gibt

172.16.0.0 bis 172.31.0.0 /16 => funktioniert nicht, da es nur 16 Netze gibt

192.168.0.0 /24 => funktioniert nicht, da es nur 256 Netze gibt

- Lösung: 10.0.0.0 /24

Nutzung des privaten Netzes aus Class A

Nutzung der Standardsubnetzmaske aus Class C

Somit ergibt sich:

1 Netz (10) aus dem Oktett 1 \* 256 Netze (0 – 255) aus dem Oktett 2 \* 256 Netze (0 – 255) aus dem Oktett 3

= 65.536 Netze (Oktette 1 bis 3 zählen durch die Subnetzmaske jetzt zur Netz-ID).

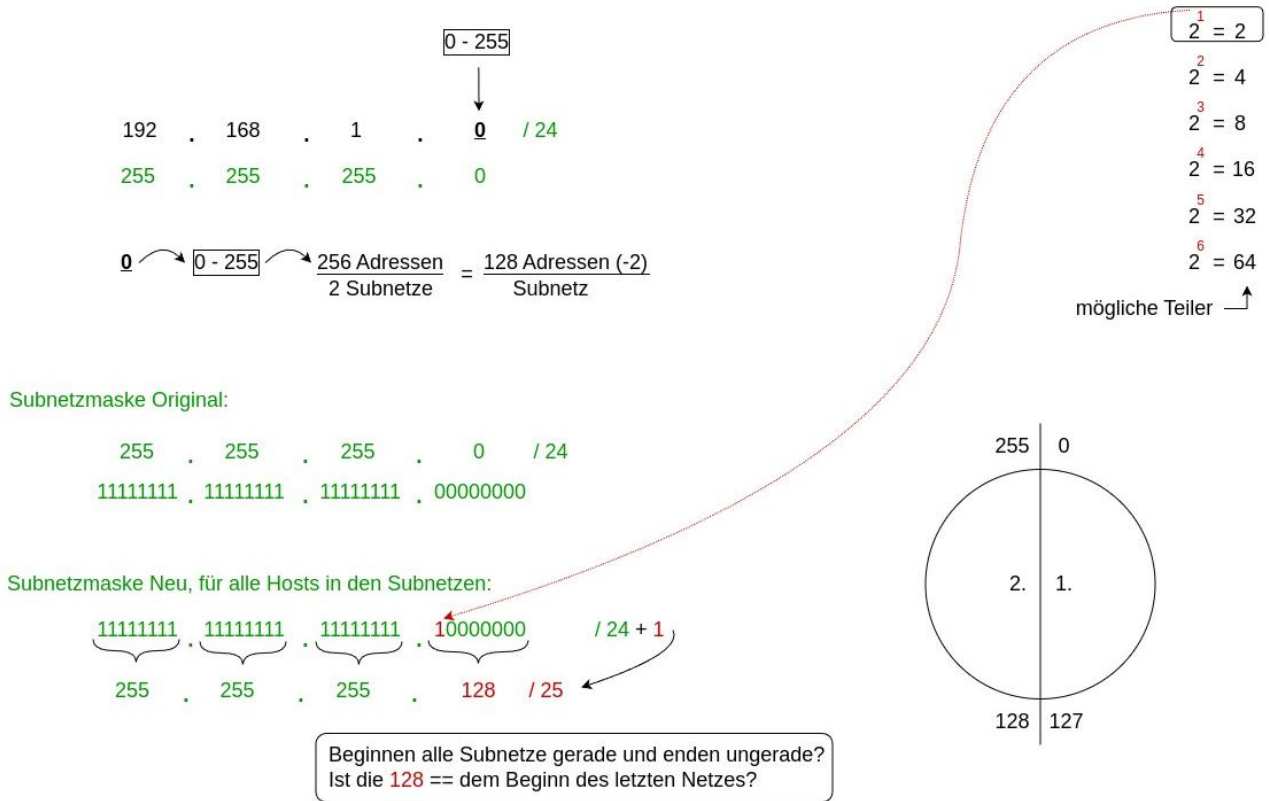
=> Jedes Netz hat 256 (-2) Adressen ( $256 = 2^8$ ) aus Oktett 4.

### 8.1.7 Subnetting IPv4 ("Richtiges Subnetting")

- Ausgegangen wird in den kommenden Beispielen vom Netz: 192.168.1.0 /24  
siehe /Filius\_Szenen/8.1.7\_LAN\_8\_PCs.flv
- benötigte Hilfsmittel:  
Umrechnung dezimal  $\Leftrightarrow$  binär  
Das „Mühlenbecker Wagenrad“  
Die Zweierpotenzen:  
 $2^1 = 2$   
 $2^2 = 4$   
 $2^3 = 8$   
 $2^4 = 16$  usw.
- Den Exponenten schreiben wir in roter Farbe!
- Der Exponent gibt an, um wie viele "Einsen" die Subnetzmaske länger wird.
- Die Zahl 2 oder 4 oder 8 oder 16 usw. ist der jeweils mögliche Teilungsfaktor:  
=> Wir können ein Netz (gleichmäßig) in 2 oder 4 oder 8 oder 16 usw. Subnetze teilen.

### 8.1.7.1 Subnetting in 2 Subnetze

siehe /Filius\_Szenen/8.1.7.1\_SUB\_2.flv



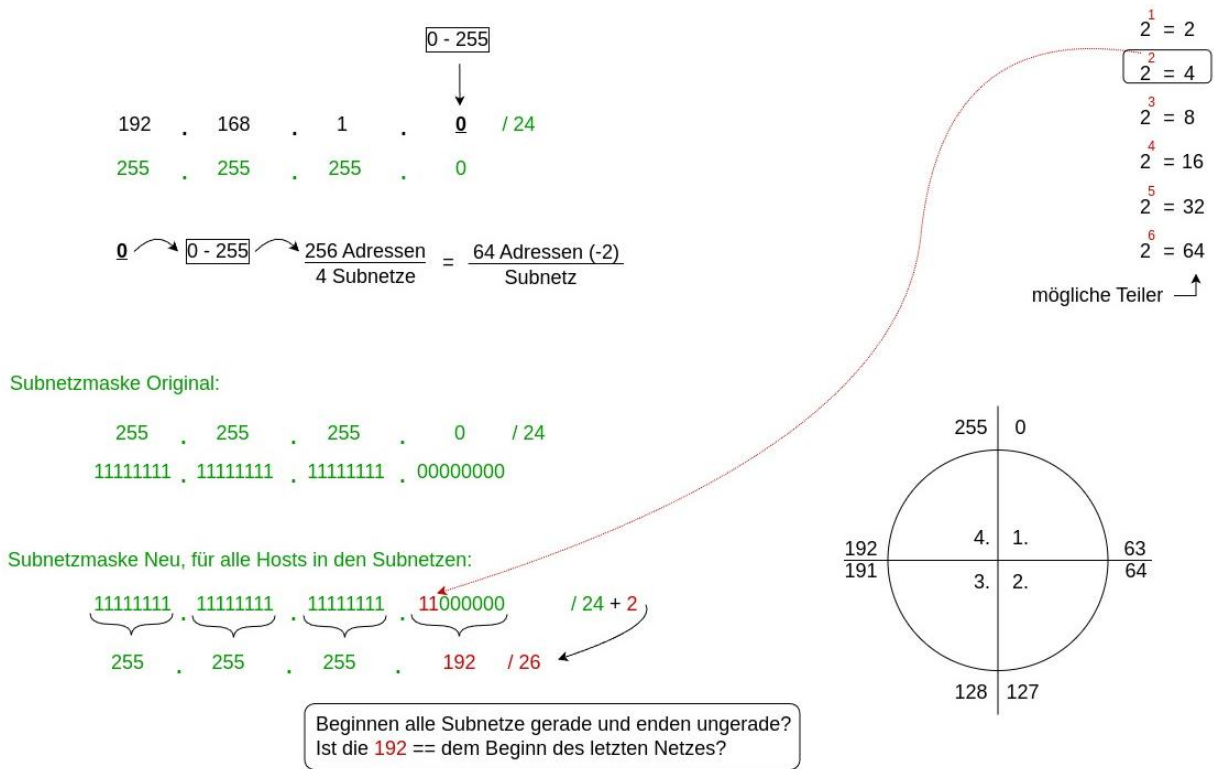
- gegeben:
  - Netz: 192.168. 1 . 0 /24 (CIDR Schreibweise)
  - Subnetzmaske: 255.255.255. 0 (dezimale Schreibweise)
- gesucht:
  - Anfang und Ende der Subnetze
  - Anzahl der Adressen pro Subnetz
  - Die neue Subnetzmaske für alle Hosts (PCs) in den jeweiligen Subnetzen

● Lösung:

- Zweierpotenzen aufschreiben, Exponent in ROT!
- Die 0 am Ende der des Netzes 192.168.1.0 als "Joker" betrachten.
- Die "Joker" 0 steht stellvertretend für alle Zahlen von 0 bis 255.
- 0 bis 255 => 256 Zahlen! (0 ist die erste Zahl).
- 256 Adressen : 2 Subnetze = 128 Adressen pro Subnetz.
- „Mühlenbecker Wagenrad“ aufzeichnen.
  - => Subnetz 1 beginnt bei 0 und endet bei 127 (= 128 Adressen).
  - => Subnetz 2 beginnt bei 128 und endet bei 255 (= 128 Adressen).
- Originale Subnetzmaske umrechnen dezimal => binär oder bei CIDR,
  - 24 Einsen schreiben und den Rest mit Nullen auffüllen:  
CIDR: /24  
dezimal: 255 . 255 . 255 . 0  
binär: 11111111.11111111.11111111.00000000
- Der kleine ROTE Exponent der Zweierpotenzen  $2^1 = 2$  besagt, die neue Subnetzmaske muss um eine Eins länger werden.
  - binär: 11111111.11111111.11111111.10000000
- Neue Subnetzmaske umrechnen binär => dezimal oder bei CIDR, Einsen zählen und aufschreiben:
  - binär: 11111111.11111111.11111111.10000000
  - dezimal: 255 . 255 . 255 . 128
  - CIDR: /25

### 8.1.7.2 Subnetting in 4 Subnetze

siehe /Filius\_Szenen/8.1.7.2\_SUB\_4.flv



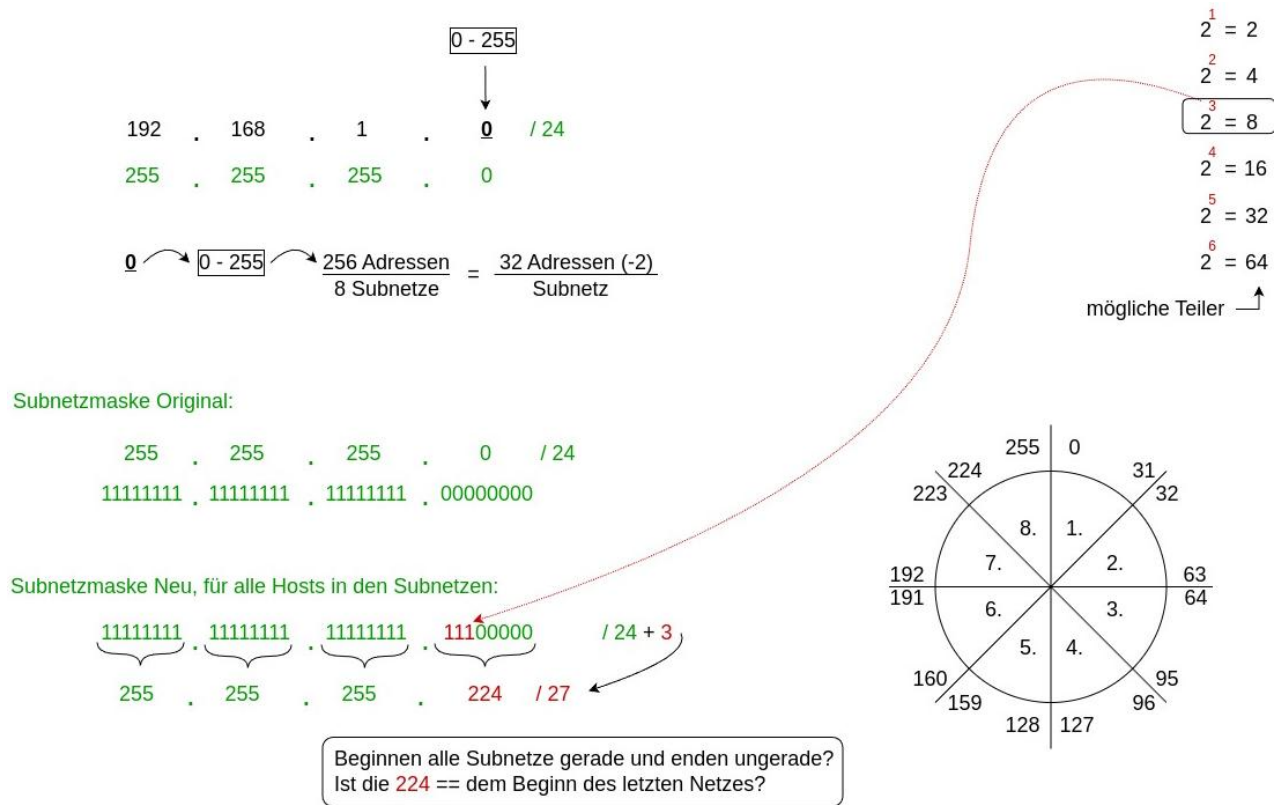
- gegeben:
  - Netz: 192.168. 1 . 0 /24 (CIDR Schreibweise)
  - Subnetzmaske: 255.255.255. 0 (dezimale Schreibweise)
- gesucht:
  - 4 Subnetze
  - Anzahl der Adressen pro Subnetz
  - Anfang und Ende der Subnetze
  - die neue Subnetzmaske für alle Hosts (PCs) in den jeweiligen Subnetzen

● Lösung

- Zweierpotenzen aufschreiben, Exponent in ROT!
- Die 0 am Ende der des Netzes 192.168.1.0 als "Joker" betrachten.
- Die "Joker" 0 steht stellvertretend für alle Zahlen von 0 bis 255.
- 0 bis 255 => 256 Zahlen! (0 ist die erste Zahl).
- 256 Adressen : 4 Subnetze = 64 Adressen pro Subnetz.
- „Mühlenbecker Wagenrad“ aufzeichnen.
  - => Subnetz 1 beginnt bei 0 und endet bei 63 (= 64 Adressen).
  - => Subnetz 2 beginnt bei 64 und endet bei 127 (= 64 Adressen).
  - => Subnetz 3 beginnt bei 128 und endet bei 191 (= 64 Adressen).
  - => Subnetz 4 beginnt bei 192 und endet bei 255 (= 64 Adressen).
- Originale Subnetzmaske umrechnen dezimal => binär oder bei CIDR
- 24 Einsen schreiben und den Rest mit Nullen auffüllen:  
CIDR: /24  
dezimal: 255 . 255 . 255 . 0  
binär: 11111111.11111111.11111111.00000000
- Der kleine ROTE Exponent der Zweierpotenzen  $2^2 = 4$  besagt, die neue Subnetzmaske muss um zwei Einsen länger werden.  
binär: 11111111.11111111.11111111.11000000
- Neue Subnetzmaske umrechnen binär => dezimal oder bei CIDR, Einsen zählen und aufschreiben:  
binär: 11111111.11111111.11111111.11000000  
dezimal: 255 . 255 . 255 . 192  
CIDR: /26

### 8.1.7.3 Subnetting in 8 Subnetze

siehe /Filius\_Szenen/8.1.7.3\_SUB\_8.flv



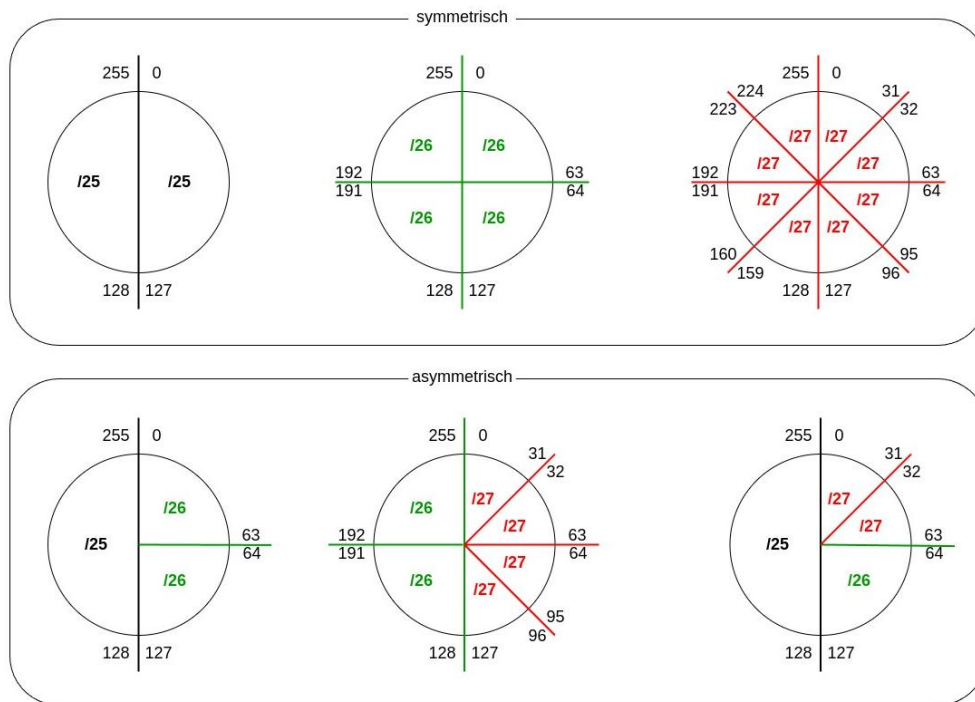
Probieren Sie es selbst:

- Wie viele Adressen sind in jedem Subnetz vorhanden?
- An welcher Adresse beginnen und enden die jeweiligen Subnetze?
- Wie lautet die neue Subnetzmaske (dezimal, CIDR)?

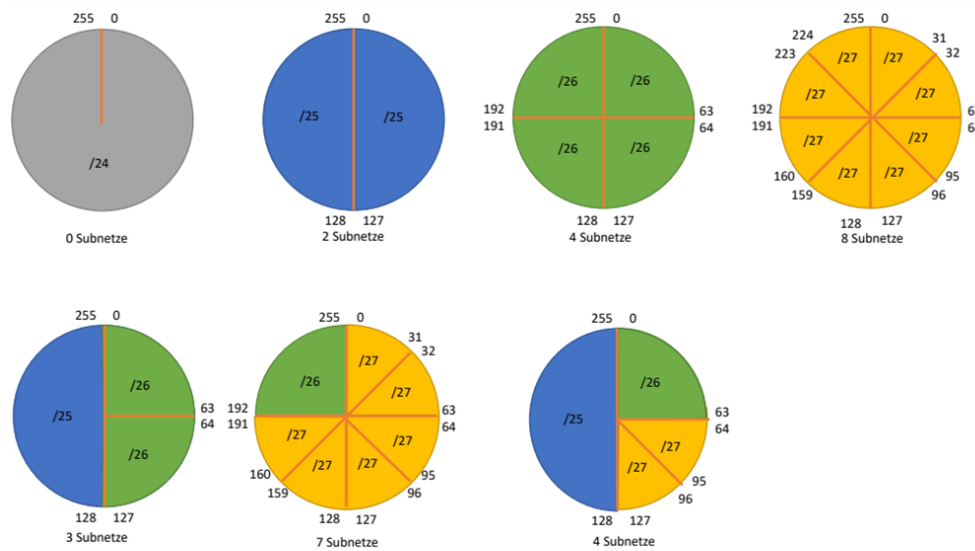
### 8.1.7.4 Asymmetrisches Subnetting (VLSM)

[de.wikipedia.org/wiki/Variable Length Subnet Mask](http://de.wikipedia.org/wiki/Variable_Length_Subnet_Mask)

siehe /Filius\_Szenen/8.1.7.4\_SUB\_asym.flv



Eine alternative Grafik eines ehemaligen Teilnehmers:



Asymmetrisches Subnetting  
Stephan Wunderlich

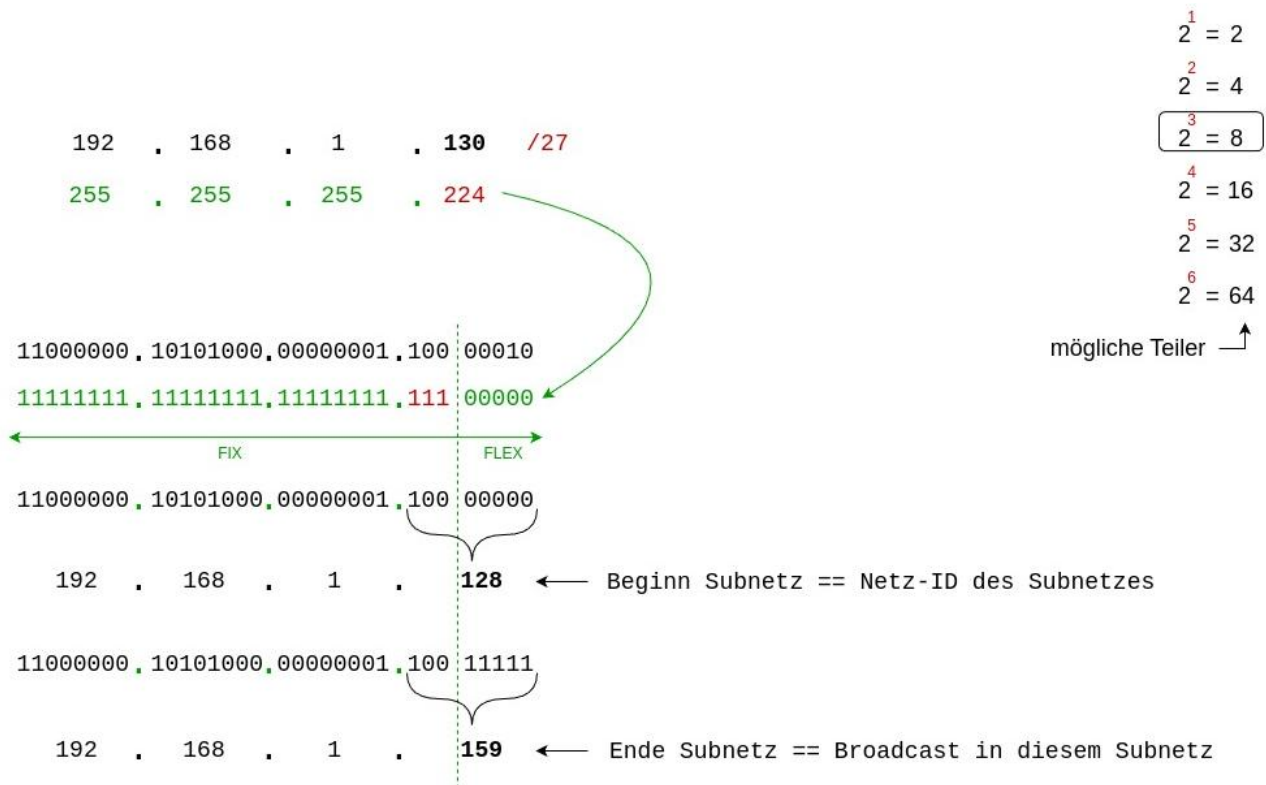
### 8.1.7.5 Subnetting Class A und Class B

Funktioniert genau so, wie in Class C beschrieben.

==>> Dabei bitte nicht die verbleibenden Oktette ("weiter rechts") vergessen!

### 8.1.8 "Reverses Subnetting" IPv4

(Anfang und Ende des Subnetzes ermitteln)



- gegeben:
  - Adresse: 192.168. 1 .130 /27 (CIDR-Schreibweise).
  - Subnetzmaske: 255.255.255.224 (dezimale Schreibweise).
- gesucht:
  - Anfang des Subnetzes, in dem sich diese IP-Adresse befindet
  - Ende des Subnetzes, in dem sich diese IP-Adresse befindet

● Lösung:

- IP-Adresse umrechnen dezimal => binär:

dezimal: 192 . 168 . 1 . 130

binär: 11000000.10101000.00000001.10000010

- Subnetzmaske umrechnen dezimal => binär oder bei CIDR
- 27 Einsen schreiben und den Rest mit Nullen auffüllen: CIDR: /27

dezimal: 255 . 255 . 255 . 224

binär: 11111111.11111111.11111111.11100000

- IP-Adresse und Subnetzmaske untereinander schreiben (binär):

IP: 11000000.10101000.00000001. 10000010

Sub: 11111111 . 11111111 . 11111111. 11100000

- Ende der "Einsen" aus der Subnetzmaske suchen und senkrechte Linie ziehen:

IP: 11000000.10101000.00000001.100 | 00010

Sub: 11111111 . 11111111 . 11111111. 111 | 00000

- Anfang des Subnetzes ermitteln:

In der IP-Adresse, rechts von der senkrechten Linie, alle bits auf 0 ("null") setzen:

IP: 11000000.10101000.00000001.100 | 00000

Jedes Oktett für sich binär => dezimal umrechnen:

IP: 11000000.10101000.00000001.100 | 00000 (binär)

IP: 192 . 168 . 1 . 128 (dezimal)

=> Vergleichen Sie die Ergebnisse mit dem „Mühlenbecker Wagenrad“

- Ende des Subnetzes ermitteln:

In der IP-Adresse, rechts von der senkrechten Linie, alle bits auf 1 ("eins") setzen:

IP: 11000000.10101000.00000001.100 | 11111

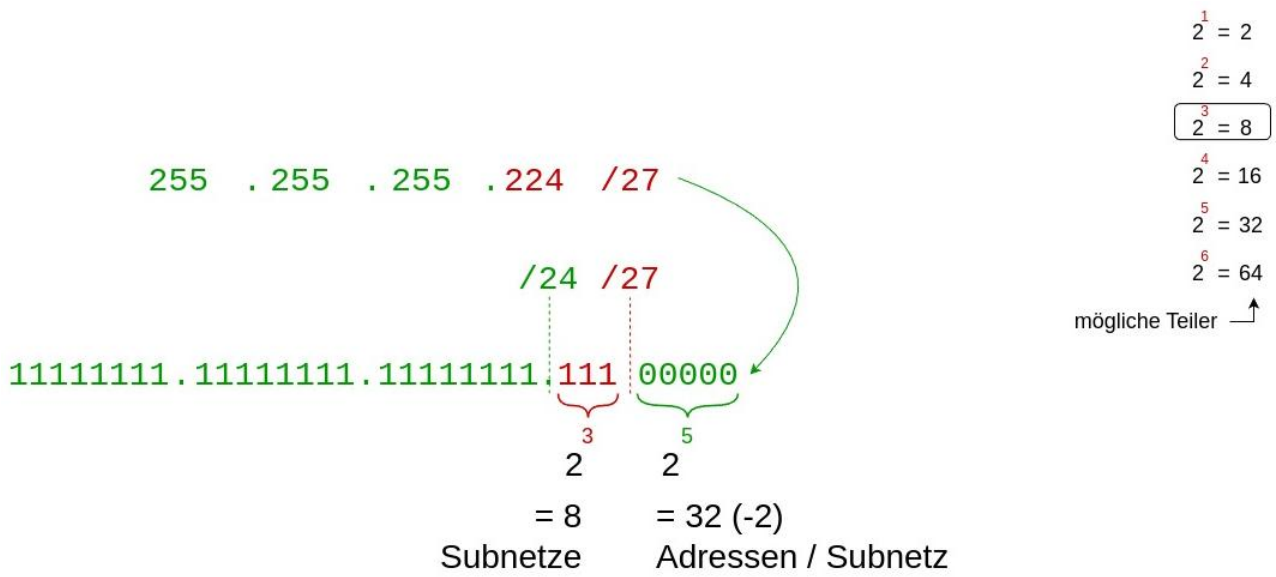
- Jedes Oktett für sich binär => dezimal umrechnen:

IP: 11000000.10101000.00000001.100 | 11111 (binär)

IP: 192 . 168 . 1 . 159 (dezimal)

=> Vergleichen Sie die Ergebnisse mit dem „Mühlenbecker Wagenrad“

### 8.1.9 Subnetzmaske IPv4 analysieren



- gegeben:
  - Subnetzmaske: 255.255.255.224 (dezimal) bzw. /27 (CIDR).
- gesucht:
  - Anzahl der Subnetze
  - Anzahl der Adressen im jeweiligen Subnetz

● Lösung:

- Subnetzmaske umrechnen dezimal => binär

dezimal: 255 . 255 . 255 . 224

binär: 11111111.11111111.11111111.11100000

- bei CIDR, 27 Einsen schreiben und den Rest mit Nullen auffüllen:
- Ende der "Einsen" aus der Standardsubnetzmaske (255.255.255.0 bzw. /24) suchen und eine senkrechte Linie ziehen:

11111111.11111111.11111111 | 11100000

- Ende der "Einsen" aus der gegebenen Subnetzmaske suchen

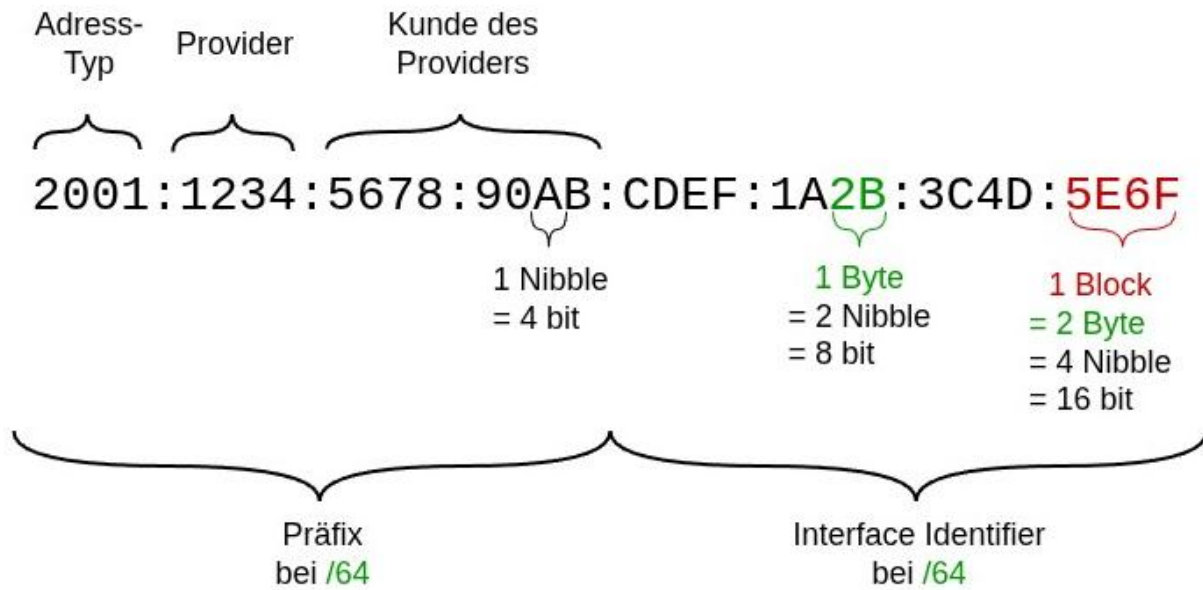
Zweite senkrechte Linie ziehen und Zahl der "Einsen" und der "Nullen" notieren:

11111111.11111111.11111111 | 111 | 00000

3 5

- $2^3$  (Einsen) = 8 => 8 Subnetze sind entstanden.
- $2^5$  (Nullen) = 32 => 32 (-2) Adressen pro Subnetz sind vorhanden

## 8.2 IPv6



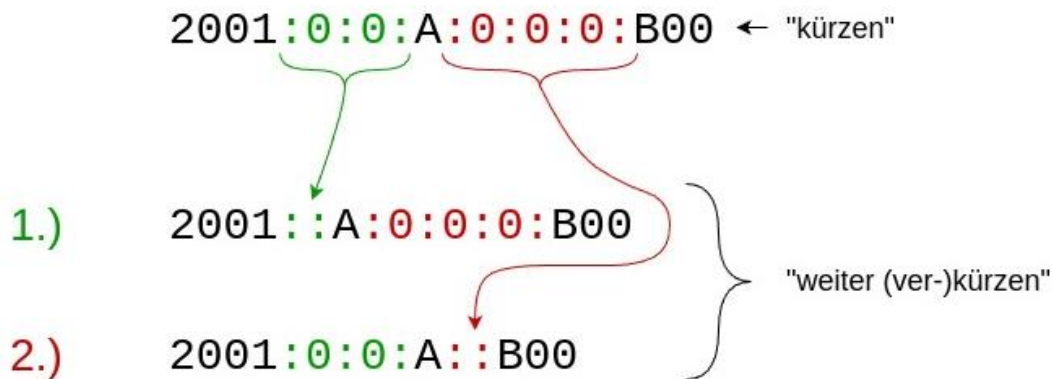
- Die IPv6-Adresse ist 128 bit lang.
- Je 4 bit werden hexadezimal als ein "Nibble" dargestellt.
- 4 "Nibble" bilden einen "Block".
- Die "Blöcke" werden durch einen Doppelpunkt : getrennt.

Somit ergeben sich:

- 128 bit
- 32 Nibble
- 16 Byte (2 Nibble = 8 bit = 1 Byte)
- 8 Blöcke
- Die Subnetzmaske wird in CIDR-Schreibweise dargestellt.
- Die Subnetzmaske muss nicht /64 sein, auch andere Werte sind möglich und üblich, z. B. /48, /56
- Der erste Block gibt Auskunft über den Typ der Adresse.
- Der zweite Block gibt Auskunft über den Provider, dem die Adresse gehört.
- Die Blöcke 3 bis 4 kennzeichnen den Kunden des Providers.

## 8.2.1 „Kürzen“ von IPv6-Adressen

2001:0000:0000:000A:0000:0000:0000:0B00



Beispiel 1, anhand der Grafik

- gegeben:
  - IPv6-Adresse: 2001:0000:0000:000A:0000:0000:0000:0B00
- gesucht:
  - Verkürzte Schreibweise der IPv6-Adresse
- Lösung:
  - Führende "Nullen" (NUR FÜHRENDE!) dürfen weggelassen werden, somit ergibt sich eine verkürzte Schreibweise der IPv6-Adresse: 2001:0:0:A:0:0:0:0B00

## 8.2.2 „Weiter verkürzen“ von IPv6-Adressen

Beispiel 1, anhand der Grafik

- gegeben:
  - verkürzte Schreibweise der IPv6-Adresse: 2001:0:0:A:0:0:0:B00
- gesucht:
  - "weiter verkürzte" Schreibweise der IPv6-Adresse.
- Lösung:
  - Einmalig (EINMALIG!) pro IPv6-Adresse darf ein Block, der nur aus "Nullen" besteht, durch zwei Doppelpunkte :: ersetzt werden.
  - Einmalig (EINMALIG!) pro IPv6-Adresse dürfen auch mehrere aufeinander folgende Blöcke, die nur aus "Nullen" bestehen, können durch zwei Doppelpunkte :: ersetzt werden.
  - Im konkreten Fall ergeben sich 2 Lösungen:
    - Lösung 1 (grüne Variante), die vorderen "Nuller-Blöcke", werden durch :: ersetzt:  
=> 2001::A:0:0:0:B00
    - Lösung 2 (rote Variante), die hinteren "Nuller-Blöcke", werden durch :: ersetzt:  
=> 2001:0:0:A::B00 => ist zu bevorzugen, da effektiver

Beispiel 2:

- gegeben:
  - IPv6-Adresse: 2001:1234:5678:90AB:0000:0000:0000:0000
- gesucht:
  - verkürzte Schreibweise der IPv6-Adresse
  - weiter verkürzte Schreibweise der IPv6-Adresse
- Lösung:
  - führende "Nullen" weglassen:
    - aus: 2001:1234:5678:90AB:0000:0000:0000:0000
    - wird: 2001:1234:5678:90AB:0:0:0:0
  - mehrere aufeinanderfolgende Blöcke, die nur aus "Nullen" bestehen, durch zwei Doppelpunkte :: ersetzen.
    - aus: 2001:1234:5678:90AB:0:0:0:0
    - wird: 2001:1234:5678:90AB::

### Beispiel 3:

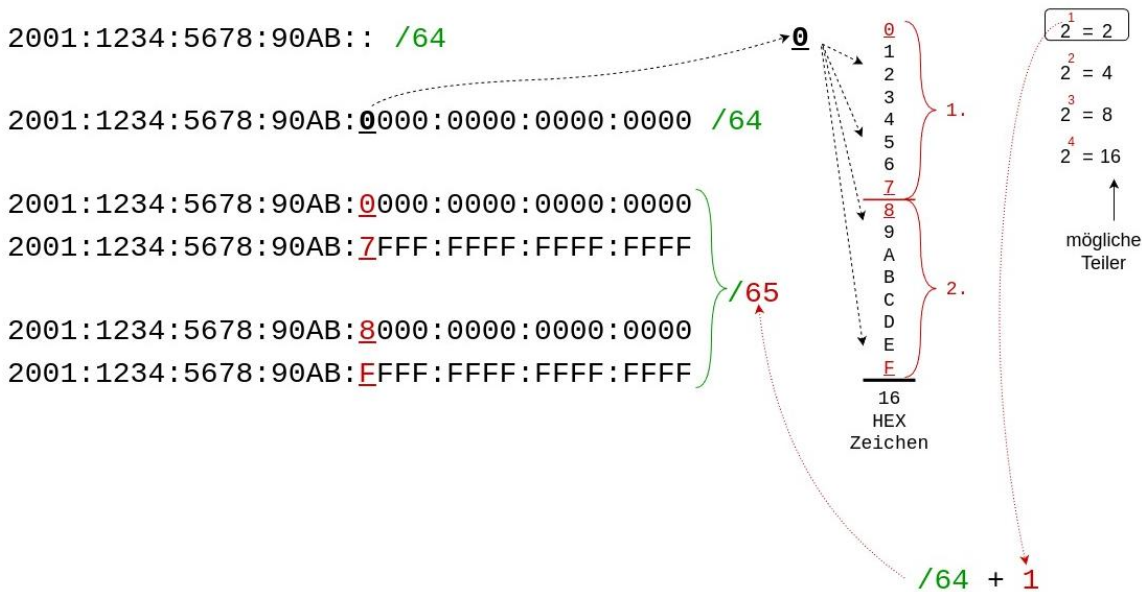
- gegeben:
  - IPv6-Adresse: 0000:0000:0000:0000:0000:0000:0000:0001
- gesucht:
  - verkürzte Schreibweise der IPv6-Adresse
  - weiter verkürzte Schreibweise der IPv6-Adresse
- Lösung:
  - führende "Nullen" weglassen:  
aus: 0000:0000:0000:0000:0000:0000:0000:0001  
wird: 0:0:0:0:0:0:0:1
  - mehrere aufeinanderfolgende Blöcke, die nur aus "Nullen" bestehen, durch zwei Doppelpunkte :: ersetzen  
aus: 0:0:0:0:0:0:0:1  
wird: ::1
  - folgende IPv6-Adressen sind somit gleichwertig:  
aus Beispiel 1:  
2001:0000:0000:000A:0000:0000:0000:0B00  
= 2001:0:0:A:0:0:0:B00  
= 2001::A:0:0:0:B00  
= 2001:0:0:A::B00  
  
aus Beispiel 2:  
2001:1234:5678:90AB:0000:0000:0000:0000  
= 2001:1234:5678:90AB:0:0:0:0  
= 2001:1234:5678:90AB::  
  
aus Beispiel 3:  
0000:0000:0000:0000:0000:0000:0000:0001  
= 0:0:0:0:0:0:0:1  
= ::1

### 8.2.3 Subnetting IPv6

Benötigte Hilfsmittel:

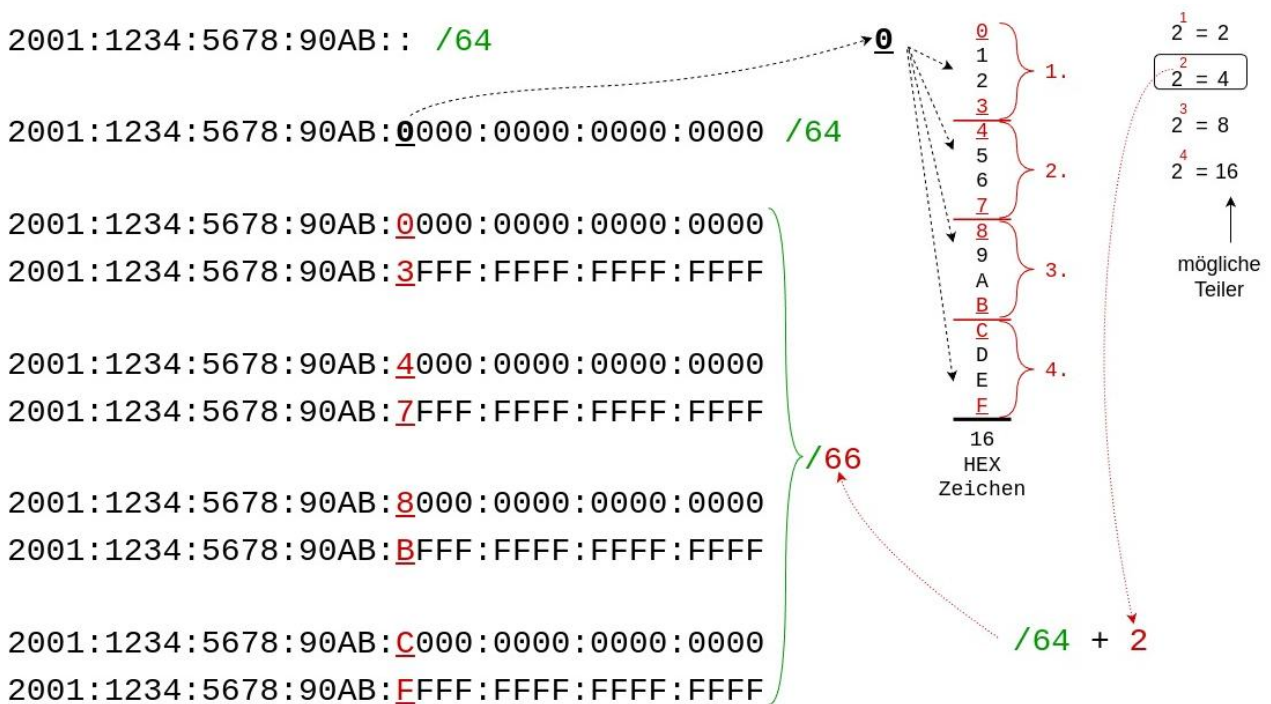
- Umrechnung hexadezimal  $\Leftrightarrow$  binär
- ----> "Mühlenbecker Pfeil": inoffizielle Darstellung, gilt NUR im BFW-Mühlenbeck! Erspart mehrere "F-Böcke" (FFFF) bis zum Ende der Adresse zu schreiben.
- – „ – "Mühlenbecker Gänsefüßchen": inoffizielle Darstellung, gilt auch NUR im BFW-Mühlenbeck! Erspart, den immer gleichen Präfix der Adresse zu schreiben.
- hexadezimale Zeichen (16 Stück): 0 1 2 3 4 5 6 7 8 9 A B C D E F
- Die Zweierpotenzen:
  - $2^1 = 2$
  - $2^2 = 4$
  - $2^3 = 8$
  - $2^4 = 16$
  - usw.
- Den kleinen Exponenten schreiben wir in roter Farbe!
- Der kleine Exponent gibt an, um wie viele "Einsen" die Subnetzmaske länger wird.
- Die Zahl 2 oder 4 oder 8 oder 16 usw. ist der jeweils mögliche Teilungsfaktor: Wir können ein Netz (gleichmäßig) in 2 oder 4 oder 8 oder 16 usw. Subnetze teilen.

### 8.2.3.1 Subnetting in 2 Subnetze



- gegeben:
  - Netz: 2001:1234:5678:90AB:: /64  
= 2001:1234:5678:90AB:0000:0000:0000:0000 /64
- gesucht:
  - 2 Subnetze
  - Anfang und Ende der Subnetze
  - neue Subnetzmaske für alle Hosts (PCs) in den Subnetzen
- Lösung:
  - Zweierpotenzen aufschreiben, Exponent in ROT!
  - neue Subnetzmaske berechnen:
  - Der kleine ROTE Exponent der Zweierpotenzen  $2^1 = 2$  besagt, die neue Subnetzmaske muss um eine Eins länger werden:  $/64 + 1 \Rightarrow /65$
  - Die erste 0 im ersten "Nuller-Block" als "Joker" betrachten.
  - Die "Joker" 0 steht stellvertretend für alle hexadezimalen Zeichen:  
0 1 2 3 4 5 6 7 8 9 A B C D E F  $\Rightarrow$  16 hexadezimalen Zeichen /  $2 = 8$   
 $\Rightarrow$  Subnetz 1 beginnt bei 0 und endet bei 7  
 $\Rightarrow$  Subnetz 2 beginnt bei 8 und endet bei F
  - Subnetze aufschreiben:
    - Subnetz 1 beginnt bei: 2001:1234:5678:90AB:0000:0000:0000:0000 /65
    - Subnetz 1 endet bei: 2001:1234:5678:90AB:7FFF:FFFF:FFFF:FFFF /65
    - Subnetz 2 beginnt bei: 2001:1234:5678:90AB:8000:0000:0000:0000 /65
    - Subnetz 2 endet bei: 2001:1234:5678:90AB:FFFF:FFFF:FFFF:FFFF /65

### 8.2.3.2 Subnetting in 4 Subnetze



● gegeben:

- Netz: 2001:1234:5678:90AB:: /64  
= 2001:1234:5678:90AB:0000:0000:0000:0000 /64

● gesucht:

- 4 Subnetze
- Anfang und Ende der Subnetze
- neue Subnetzmaske für alle Hosts (PCs) in den Subnetzen

● Lösung:

- Zweierpotenzen aufschreiben, Exponent in ROT!
- neue Subnetzmaske berechnen:
- Der kleine ROTE Exponent der Zweierpotenzen  $2^2 = 4$  besagt, die neue Subnetzmaske muss um zwei Einsen länger werden:  $/64 + 2 \Rightarrow /66$
- Die erste 0 im ersten "Nuller-Block" als "Joker" betrachten.
- Die "Joker" 0 steht stellvertretend für alle hexadezimalen Zeichen:  
0 1 2 3 4 5 6 7 8 9 A B C D E F  $\Rightarrow$  16 hexadezimalen Zeichen / 4 = 4  
 $\Rightarrow$  Subnetz 1 beginnt bei 0 und endet bei 3  
 $\Rightarrow$  Subnetz 2 beginnt bei 4 und endet bei 7  
 $\Rightarrow$  Subnetz 3 beginnt bei 8 und endet bei B  
 $\Rightarrow$  Subnetz 4 beginnt bei C und endet bei F

- Subnetze aufschreiben:

Subnetz 1 beginnt bei: 2001:1234:5678:90AB:0000:0000:0000:0000 /66

Subnetz 1 endet bei: 2001:1234:5678:90AB:3FFF:FFFF:FFFF:FFFF /66

Subnetz 2 beginnt bei: 2001:1234:5678:90AB:4000:0000:0000:0000 /66

Subnetz 2 endet bei: 2001:1234:5678:90AB:7FFF:FFFF:FFFF:FFFF /66

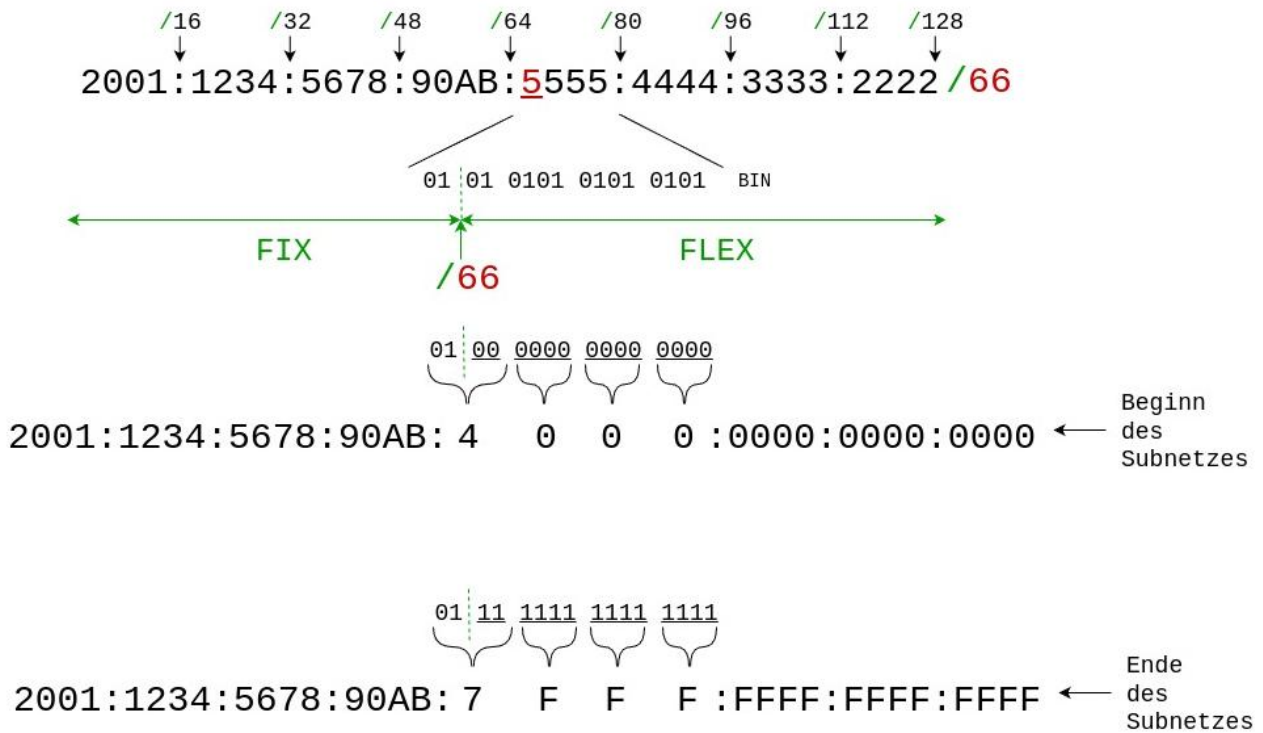
Subnetz 3 beginnt bei: 2001:1234:5678:90AB:8000:0000:0000:0000 /66

Subnetz 3 endet bei: 2001:1234:5678:90AB:BFFF:FFFF:FFFF:FFFF /66

Subnetz 4 beginnt bei: 2001:1234:5678:90AB:C000:0000:0000:0000 /66

Subnetz 4 endet bei: 2001:1234:5678:90AB:FFFF:FFFF:FFFF:FFFF /66

### 8.2.4 "Reverses Subnetting" IPv6



- gegeben:
  - Adresse: 2001:1234:5678:90AB:5555:4444:3333:2222
  - Subnetzmaske: /66
  
- gesucht:
  - Anfang des Subnetzes, in dem sich diese IP-Adresse befindet
  - Ende des Subnetzes, in dem sich diese IP-Adresse befindet

● Lösung:

- Die Stelle in der Adresse suchen, an der sich das 66. bit befindet:

1 Block = 16 bit => Block 5 beinhaltet die bits 65 bis 80

- Den Block 5 "zerlegen" (hexadezimal => binär)

66 bits abzählen (64 + 2) und eine senkrechte Linie ziehen

- Anfang des Subnetzes ermitteln:

- Alle bits nach dem 66. bit (rechts von der senkrechten Linie) auf 0 (binär) setzen

- Jedes Nibble aus Block 5 für sich binär => hexadezimal zurückrechnen.

- Adresse wieder vollständig aufschreiben und dabei an die restlichen Blöcke denken, die jetzt zu "Nuller-Blöcken" geworden sind

- Ende des Subnetzes ermitteln:

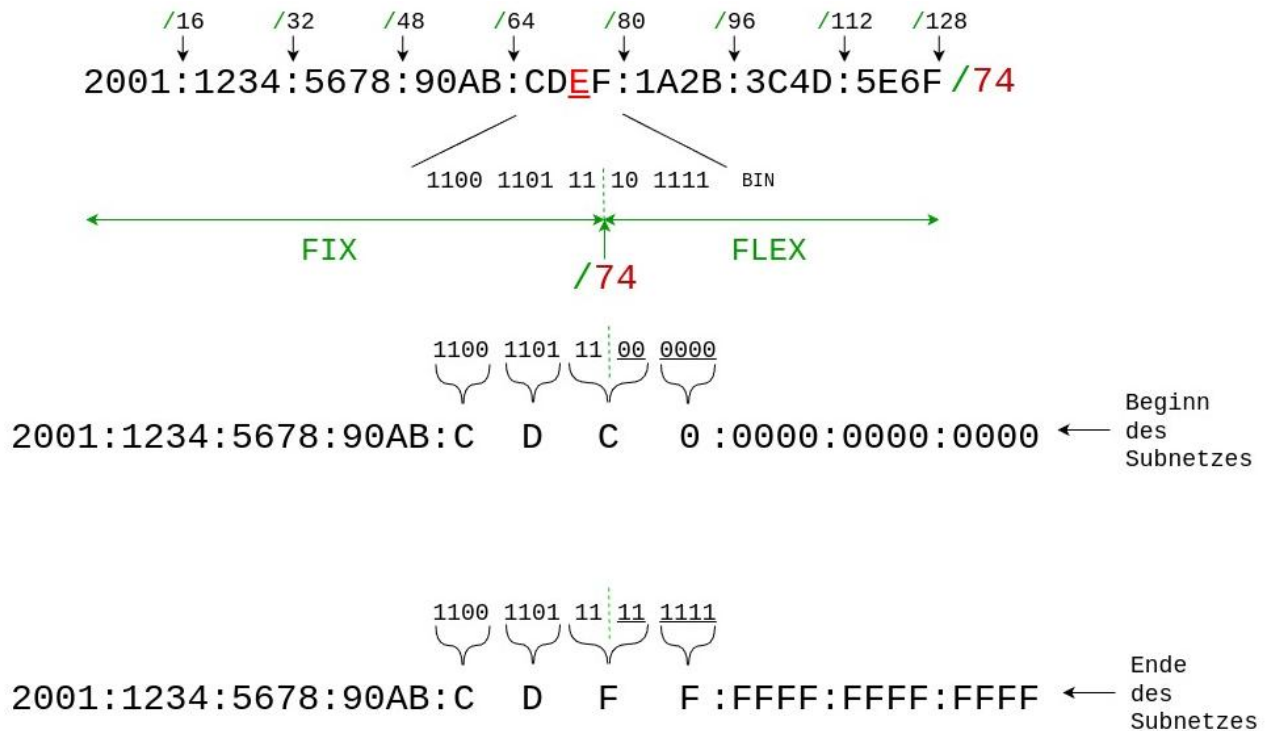
- Alle bits nach dem 66. bit (rechts von der senkrechten Linie) auf 1 (binär) setzen

- Jedes Nibble aus Block 5 für sich binär => hexadezimal zurückrechnen.

- Adresse wieder vollständig aufschreiben und dabei an die restlichen Blöcke denken, die jetzt zu "FFFF-Blöcken" geworden sind.

- => Vergleichen Sie das Ergebnis mit dem IPv6-Subnetting in 4 Subnetze.

Die folgende Grafik zeigt eine Variante, die auch in der Prüfung vorkommen kann.



## 8.2.5 Besondere IPv6-Adressen (Auszug)

[de.wikipedia.org/wiki/IPv6](https://de.wikipedia.org/wiki/IPv6)

[heise.de/IPv6-Adressen-3484199](https://heise.de/IPv6-Adressen-3484199)

### 8.2.5.1 Nicht spezifizierte IPv6-Adresse

- 0000:0000:0000:0000:0000:0000:0000:0000  
= 0:0:0:0:0:0:0:0  
= ::

entspricht der IPv4-Adresse: 0.0.0.0

### 8.2.5.2 localhost / loopback-Adresse

- 0000:0000:0000:0000:0000:0000:0000:0001  
= 0:0:0:0:0:0:0:1  
= ::1

entspricht der IPv4-Adresse: 127.0.0.1

### 8.2.5.3 Global Unicast

2000:: /3

- liegt zwischen: 2000:0000:0000:0000:0000:0000:0000:0000 /3  
und: 3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF /3  
im Internet gültiger Adressbereich

### 8.2.5.4 Unique Local Unicast

FC00:: /7

- liegt zwischen: FC00:0000:0000:0000:0000:0000:0000:0000 /7  
und: FDFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF /7
    - Private IPv6-Adressen
    - dürfen das Unternehmensnetz nicht verlassen
    - sind nicht im Internet gültig.
- => siehe private IPv4-Adressen.

### 8.2.5.5 Link Local Unicast

FE80:: /10

- liegt zwischen: FE80:0000:0000:0000:0000:0000:0000:0000 /10  
und: FEBF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF /10  
zur Zeit genutzt (Festlegung): von FE80:: bis FE80::FFFF:FFFF:FFFF:FFFF /10  
Wird genutzt für:  
Autokonfiguration  
Neighbor-Discovery
- dürfen das Netzsegment nicht verlassen
- keine vergleichbare IPv4-Adresse (erinnert ein wenig an IPv4 APIPA)

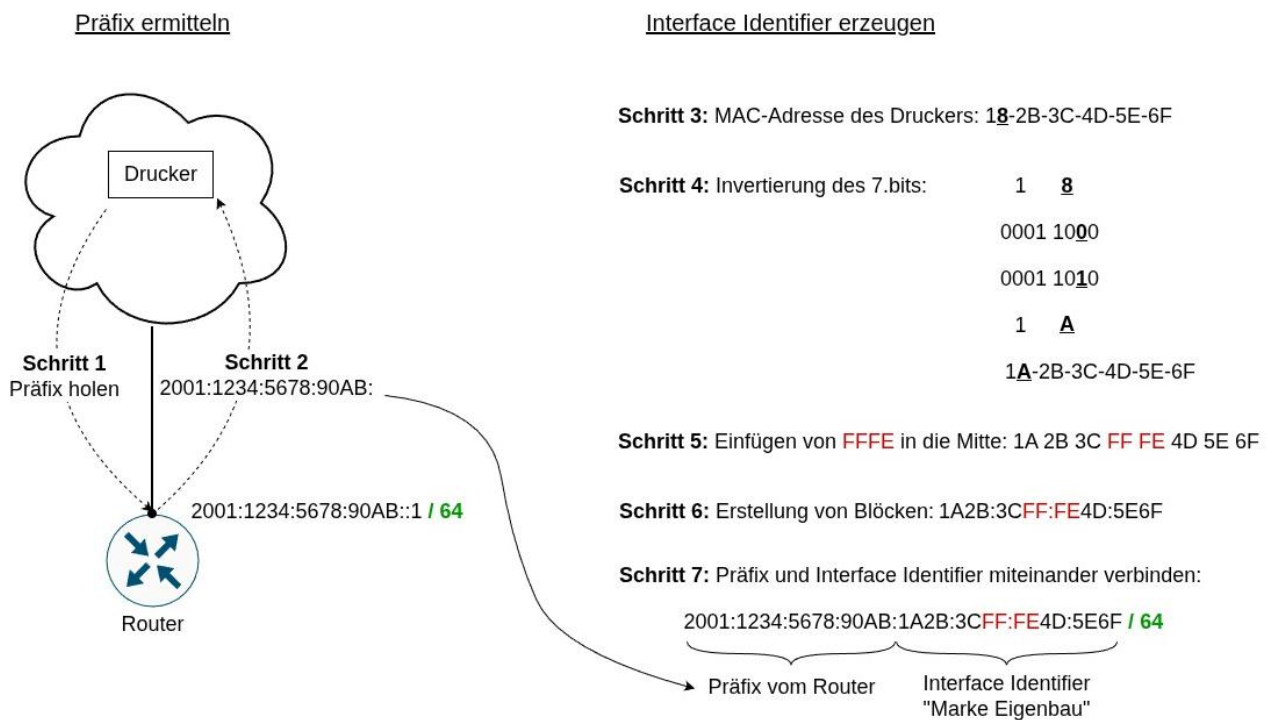
### 8.2.5.6 Multicast

[de.wikipedia.org/wiki/Multicast](http://de.wikipedia.org/wiki/Multicast)

- FF00:: /8
- (FFXY:: /8)  
liegt theoretisch zwischen: FF00:0000:0000:0000:0000:0000:0000:0000 /8  
und: FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF /8
- ist der Ersatz für IPv4-Broadcast
- Bedeutung des "X" nach FF: 4 bits für Flags (Zustandsanzeigen)
- Bedeutung des "Y" nach FFX: 4 bits für den Gültigkeitsbereich (bis wohin gilt dieser Multicast)

## 8.2.6 SLAAC - Stateless Address Autoconfiguration

„Eigenbau“ einer IPv6-Adresse, wenn kein DHCP-Server vorhanden ist => eine Art „IPv6-APIPA“



Ablauf:

- IPv6-Präfix vom Router holen.

- Interface Identifier selbst basteln:

Eigene MAC-Adresse (48 bit) nutzen und das 7. bit negieren.

=> reale Netzwerkkarten-MAC => virtuelle Netzwerkkarten-MAC

In die Mitte der MAC-Adresse wird immer der String „FFFE“ eingefügt: aus 48 bit => 64 bit

Aufteilung in Blöcke

IPv6-Präfix mit Interface Identifier miteinander verbinden

=> IPv6-Adresse, passend zum Netzsegment und nicht wie bei IPv4, die APIPA-Adresse, die immer 169.254.x.y ist

## 8.3 Vergleich der Header von IPv4 und IPv6

siehe Westermann, Seite 582

### 8.3.1 IPv4-Header

- variable Länge
- Header-Länge = Wert in Feld (2), muss mit 32 bit multipliziert werden
- In der Darstellung auf Seite 314
- Wert in Feld (2) => je eine Zeile
- Sollte ein "Universal"-Header werden (wie ein behördlicher Universalvordruck)
- Viele Optionen sind schon standardmäßig im Header vorhanden.
- Der Header ist um weitere Optionen erweiterbar.
- Der IPv4-Header hat im Gegensatz zum IPv6-Header eine Prüfsumme.
- Das Feld TTL: siehe /Filius\_Szenen/8.3.1\_Traceroute.flv

### 8.3.2 IPv6 Header

- hat eine konstante Länge
- kein "Universal"-Header
- um weitere Optionen durch "Extension"-Header erweiterbar (siehe "Next Header")
- Header hat KEINE Prüfsumme => ein Schelm, wer Böses dabei denkt
- hat ein neues Feld "Traffic Class" => Priorisierung: von: binär 00000000 => "wenn mal Zeit ist", bis: binär 11111111 => "Blaulicht mit Martinshorn"
- hat ein neues Feld "Flow Label": Alle Datenpakete, die zu einer Sitzung gehören (z. B. VoIP), bekommen das gleiche Label. Super, um zu sniffen => alle zusammengehörigen Datenpakete haben das gleiche Label

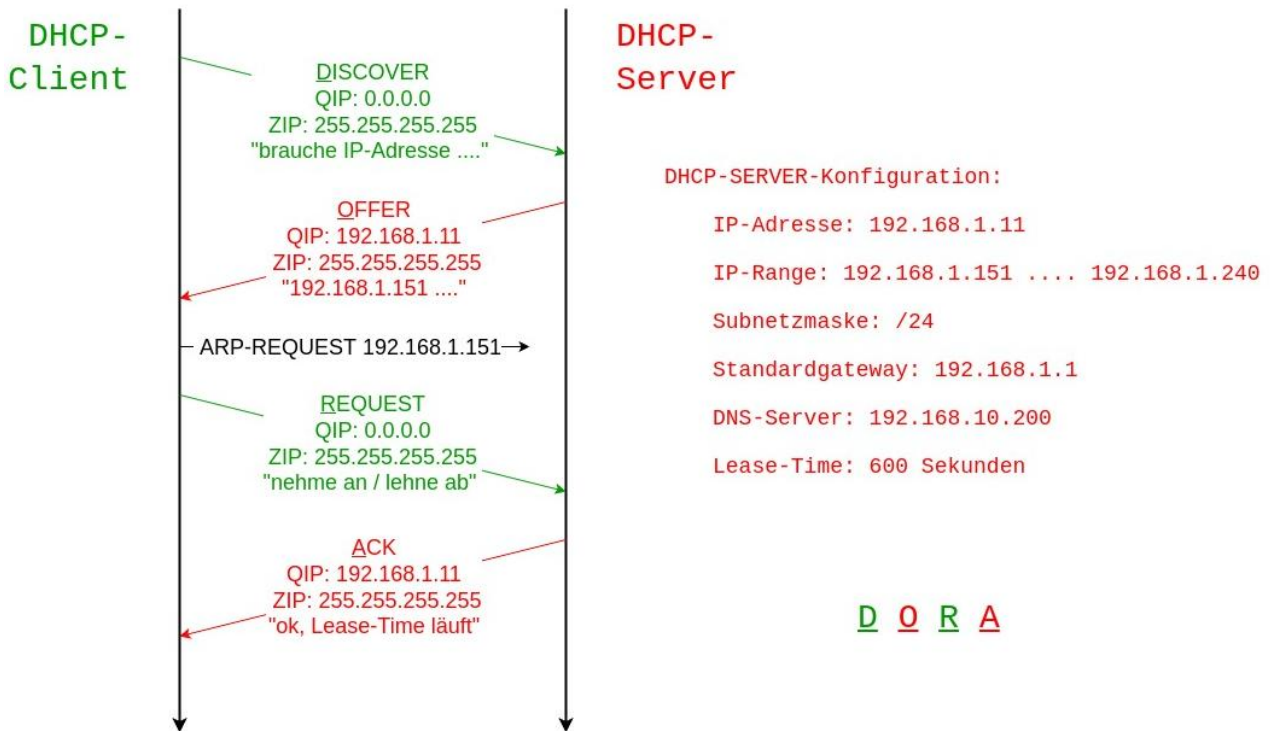
## 8.4 DHCP

siehe:

[de.wikipedia.org/wiki/Dynamic\\_Host\\_Configuration\\_Protocol](https://de.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol)

/Filius\_Szenen/8.4\_DHCP\_statisch.flv

Westermann, Seite 595



Eine Lösung, um PCs automatisch folgende Informationen zu geben (Auszug):

- IP-Adresse, Subnetzmaske, Standard-Gateway, DNS-Server
- Lease-Time (Gültigkeitsdauer dieser Angaben)

### 8.4.1 Allgemeine Erklärung zum Ablauf

siehe /Filius\_Szenen/8.4.1\_DHCP\_allgemein.fls

- DHCP-Client startet "DISCOVER":
- Quell-IP: 0.0.0.0 ("Ich weiß nicht, wer ich bin.")
- Ziel-IP: 255.255.255.255 ("Mega-Broadcast", Hilferuf an jeden). „Ich brauche alle Informationen zu den Netzwerkeinstellungen“.
- DHCP-Server unterbreitet ein "OFFER" (ein Angebot aus dem DHCP-Pool):  
IP-Adresse, Subnetzmaske, Standard-Gateway, DNS-Server, Lease-Time
- DHCP-Client überprüft die Offerte mit einem ARP-Request, ob die angebotene IP-Adresse überhaupt noch frei ist:  
wenn ja => DHCP-Client antwortet mit "REQUEST" (nehme OFFER an)
- DHCP-Server sendet ein "ACK", Lease-Time läuft ab jetzt  
wenn nein => DHCP-Client antwortet mit "DECLINE" (lehne OFFER ab) der DHCP-Client startet erneut ein "Discover"

siehe /Filius\_Szenen/8.4.1\_DHCP\_Konflikt.fls

- Die Anfangsbuchstaben der 4 "Pfeile" (Discover, Offer, Request, ACK), ergeben den weiblichen Vornamen: "DORA".
- 4 "Pfeile" => 4 Datenpakete sind für DHCP nötig: => 4-Wege-Handshake-Verfahren.

### 8.4.2 Tipps zu DHCP

- Statische Hosts (Server, Drucker, PCs) bekommen statische IPs => viel einfachere Fehlersuche!
- Flexible Hosts (Laptops der Außendienstmitarbeiter) nutzen DHCP.

## 8.5 Namensauflösung

[de.wikipedia.org/wiki/Domain \(Internet\)#Fully Qualified Domain Name \(FQDN\)](https://de.wikipedia.org/wiki/Domain_(Internet)#Fully_Qualified_Domain_Name_(FQDN))

- Warum Namensauflösung: Hosts (Computer, Server) bekommen einen Namen, da sich die meisten Menschen IP-Adressen (IPv4, IPv6) schlecht merken können.

Beispiel:

nslookup www.heise.de

Address: 193.99.144.85 <= IPv4-Adresse

Address: 2a02:2e0:3fe:1001:7777:772e:2:85 <= IPv6-Adresse

- Ein Host (Computer, Server) hat IMMER einen Netbios-Namen.
- Ein Host (Computer, Server) hat OPTIONAL einen DNS-Namen (FQDN).

### 8.5.1 Netbios-Name

- NICHT für Anfragen aus dem Internet zu gebrauchen, nur für das interne Netz (Broadcastdomain)
- maximal 15 vergebare Zeichen (Bitte nur folgende Zeichen verwenden):
  - a...z
  - A...Z
  - 0...9
  - (Minuszeichen)

Beispiel: Server-1

- Wie funktioniert die Namensauflösung (Netbios-Name => IP-Adresse) am Beispiel Server-1?
  - per Broadcast ("Wer ist Server-1, ich brauche deine IP-Adresse")
  - per Datei "lmhosts" (völlig veraltet)
  - per Dienst "WINS" (völlig veraltet)

## 8.5.2 DNS-Name (FQDN)

siehe Westermann, Seite 595

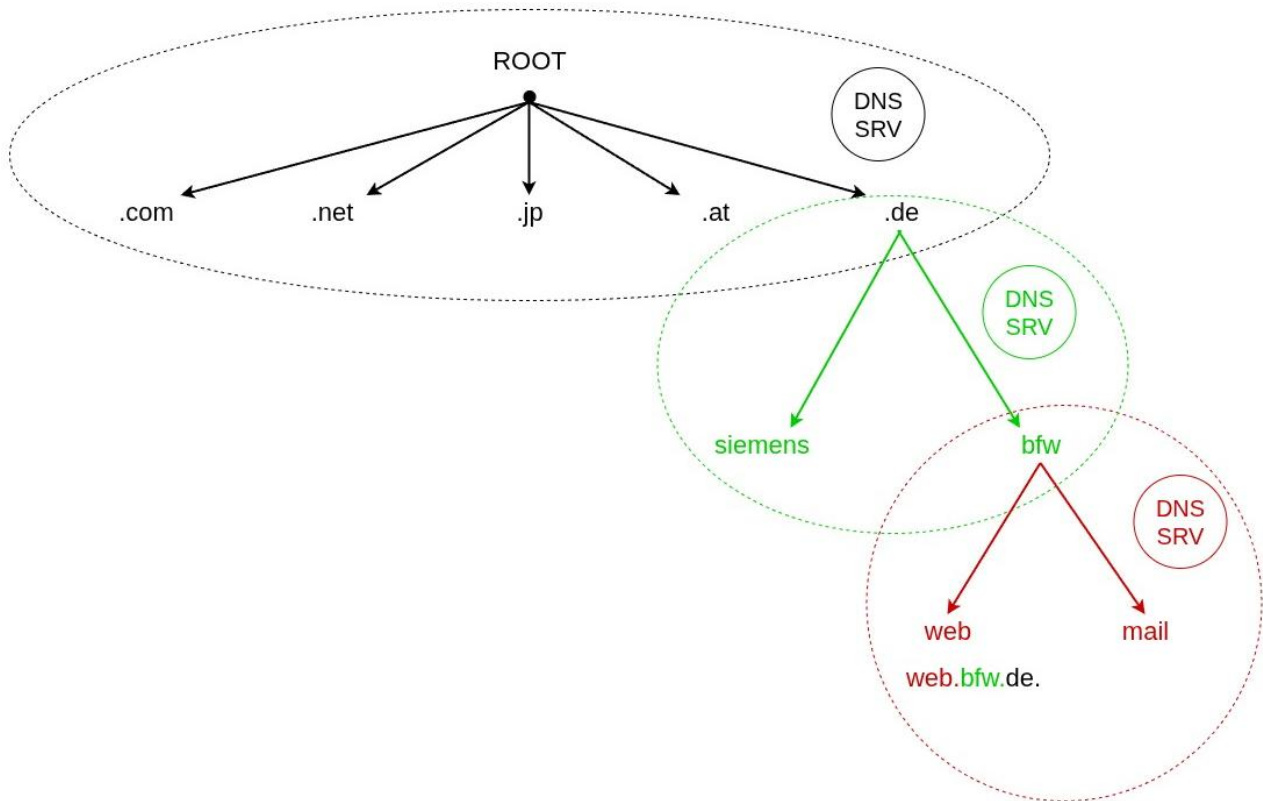
siehe /Filius\_Szenen/8.5.2\_DNS\_einfach.flv

siehe /Filius\_Szenen/8.5.2\_Home\_Berlin\_Paris\_DNS\_einfach.flv

siehe /Filius\_Szenen/8.5.2\_Home\_Berlin\_Paris\_DNS\_komplex.flv

- Wird gebraucht, wenn der Host (Computer, Server) aus dem Internet oder außerhalb der Broadcast-Domain erreichbar sein soll.
- Wird auch gebraucht, wenn eine Windows-Domäne aufgebaut werden soll.
- Maximal 255 vergebare Zeichen, nach 63 Zeichen muss ein „.“ (Punkt) gesetzt werden. Bitte nur das englische Alphabet verwenden (kann sonst zu Problemen kommen, wie soll jemand deutsche Umlaute eingeben, der ä, ö, ü, ß nicht auf seiner Tastatur hat).

### 8.5.2.1 Die Struktur von DNS



### 8.5.2.2 Praktische Lösung für Zugriff aus dem Internet

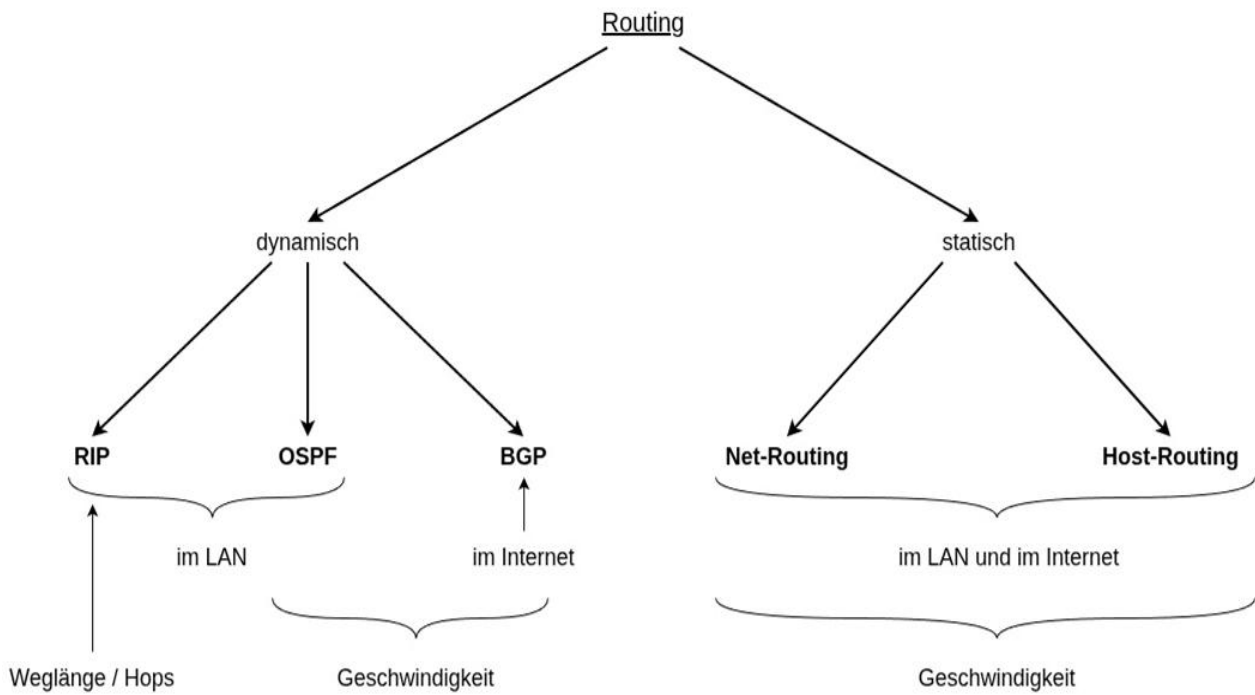
- Domain registrieren (z. B. bfw.de).
- Dem Server einen Netbios-Namen geben (z. B. Server-1).
- Netbios-Namen und Domain verbinden: Server-1.bfw.de
- Da niemand im Internet wissen kann, wie der Netbios-Name des Servers lautet, wird der Alias „www“ verwendet. (Wie im richtigen Leben, man kennt nicht den Vornamen einer Person und sagt daher „Frau“ oder „Herr“)  
=> der Server ist aus dem Internet erreichbar unter: [www.bfw.de](http://www.bfw.de)

### 8.5.2.3 Praktische Lösung für interne Namensauflösung

- Domain ausdenken (z. B. itm.intern).
- Dem Server einen sinnvollen Netbios-Namen geben (z. B. proxy).
- Netbios-Namen und Domain verbinden: proxy.itm.intern

## 8.6 Routing

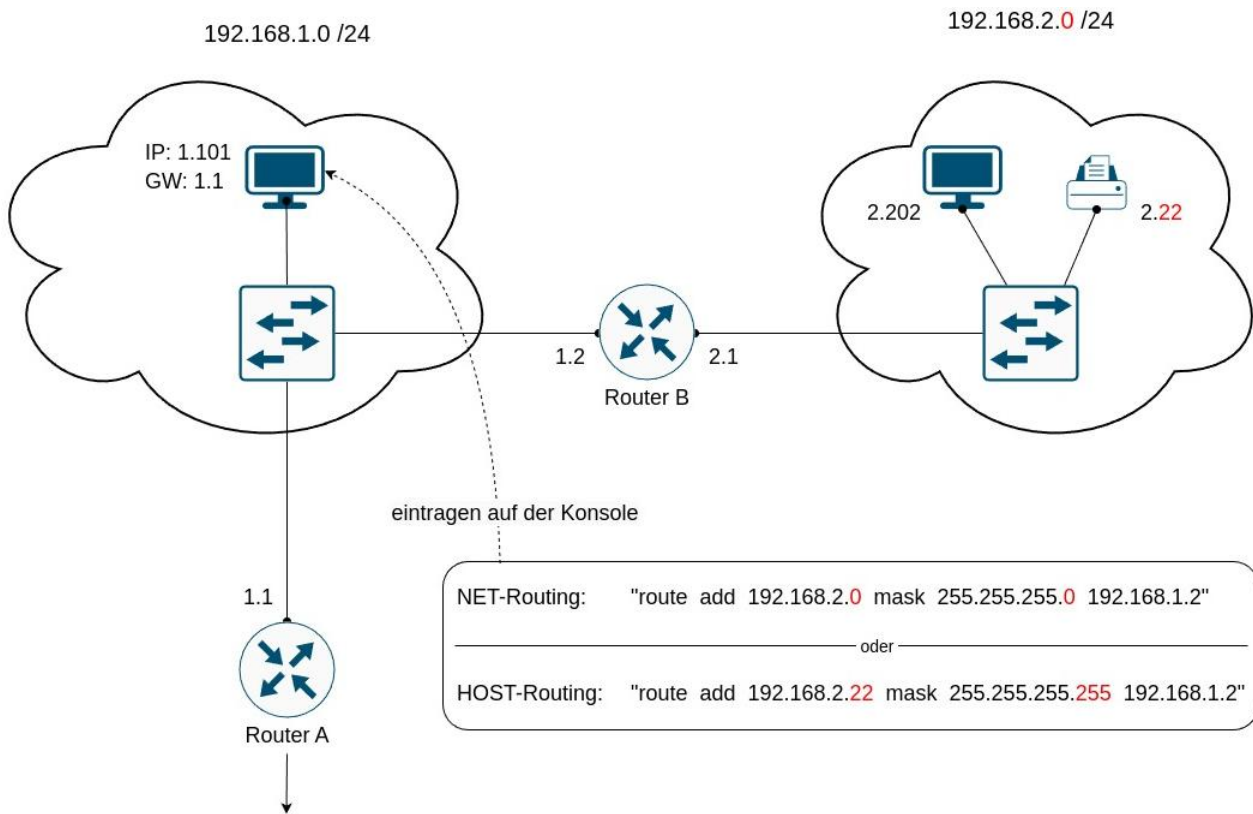
siehe Westermann Seite 603



### 8.6.1 Dynamisches Routing im LAN (Navi im Auto)

- Kürzeste Strecke (egal wie schnell), siehe Protokoll RIP  
siehe /Filius\_Szenen/8.6.1\_RIP.flv
- Schnellste Strecke (egal wie weit), siehe Protokoll OSPF

## 8.6.2 Statisches Routing ("Ich fahre immer dort entlang")



- Standard-Gateway: 192.168.1.1
- Alle Datenpakete, die nicht zum eigenen Netz gehören, werden an diese Adresse geschickt => niemand würde das Netz 192.168.2.0 oder den Computer 192.168.2.202 erreichen können.
- Lösung: statische Einträge, um die Erreichbarkeit zu gewährleisten.

### 8.6.2.1 "Net-Routing" 192.168.2.0 (Syntax Windows)

- "route add 192.168.2.0 mask 255.255.255.0 192.168.1.2"
- menschlich ausgedrückt: Wenn du ein Datenpaket für irgendjemanden im Netz 192.168.2.0 hast, dann übergib es dem Router 192.168.1.2.
- Beim "Net-Routing" verlangt die Syntax die Angabe des Netzes mit der 0 "Null" am Ende und eine "normale" Subnetzmaske (255.255.255.0).

=> Aus dem Netz 192.168.1.0 sind alle PCs der anderen Netze erreichbar.

- ein etwas komplexeres Net-Routing, siehe /Filius\_Szenen/8.6.2.1\_Net\_Routing.flv
- Vorsicht Loop bei unbekannter IP-Adresse! siehe /Filius\_Szenen/8.6.2.1\_Routing\_GW.flv

### 8.6.2.2 "Host-Routing" 192.168.2.22 (Syntax Windows)

- "route add 192.168.2.22 mask 255.255.255.255 192.168.1.2"
- menschlich ausgedrückt: Wenn du ein Datenpaket ausschließlich für den Host 192.168.2.22 hast, dann übergib es dem Router 192.168.1.2
- Beim "Host-Routing" verlangt die Syntax die Angabe des Hosts (PCs) mit der konkreten IP-Adresse am Ende und einer "speziellen" Subnetzmaske: 255.255.255.255 oder CIDR /32

=> vom PC 192.168.1.101 ist jetzt nur der Drucker 192.168.2.22 erreichbar

- Soll in diesem Beispiel auch noch der Host 192.168.2.202 erreicht werden, müsste auch für diesen Host ein neues, eigenes Host-Routing erfolgen:

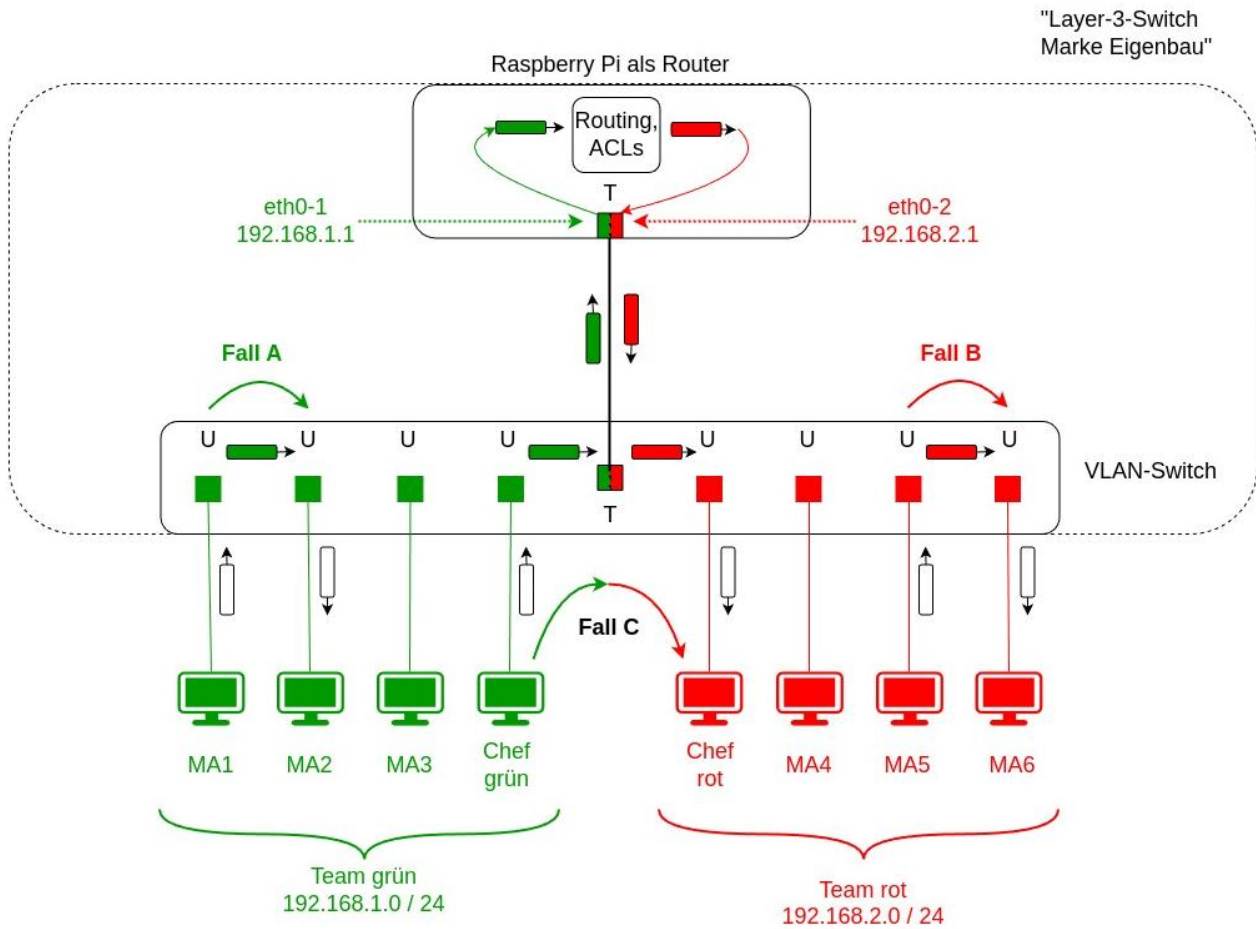
```
"route add 192.168.2.202 mask 255.255.255.255 192.168.1.2"
```

=> Vom PC 192.168.1.101 wären dann erreichbar:

- Drucker 192.168.2.22
- PC 192.168.2.202
- ein etwas komplexeres Host-Routing siehe /Filius\_Szenen/8.6.2.2\_Host\_Routing.flv

## 8.7 Layer-3-Switch

siehe /Filius\_Szenen/8.7\_Layer-3-Switch\_schummel.fls



- ein Switch, der VLANs bilden kann und zwischen den VLANs routet  
es wurden in diesem Beispiel 2 VLANs gebildet (grün und rot)
- In den Fällen „A“ und „B“ erkennt man einen klassischen VLAN-Switch => ein Datenaustausch kann nur unter den Mitarbeitern der jeweiligen Teams erfolgen.
- im Fall „C“ bleibt das grüne TAG erhalten und wird im Router in ein rotes TAG geändert => das Datenpaket kann die Grenze zwischen den VLANs überwinden

==>> äußerst moderne Form der Segmentierung eines LANs

## 9 Schicht Vier

[de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell](https://de.wikipedia.org/wiki/Internetprotokollfamilie#TCP/IP-Referenzmodell)

Themen aus Schicht 4:

- Ports
- TCP:
  - Verbindungsaufbau
  - Verbindungsabbau
  - Sliding Window
  - TCP-Header
- UDP:
  - UDP-Header
- Portknocking
- Portforwarding / Destination NAT
- NAT (PAT) / Source NAT
- Black- und Whitelist (Block- und Allowlist)

### 9.1 Ports

[iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt](https://iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt)

- Ports werden von TCP und UDP genutzt, siehe Westermann Seite 595:
- Ports befinden sich zwischen der Schicht 4 (TCP bzw. UDP) und den Anwendungsschichten 5 bis 7.
- Denken Sie an den Abfluss eines Waschbeckens:
  - Normalerweise läuft das Wasser von der Anwendung (Hände waschen) in das darunterliegende System ab.
  - Bekanntlich kann es auch passieren, dass das Wasser von unten in das Waschbecken zurückgedrückt wird (nicht wirklich angenehm).

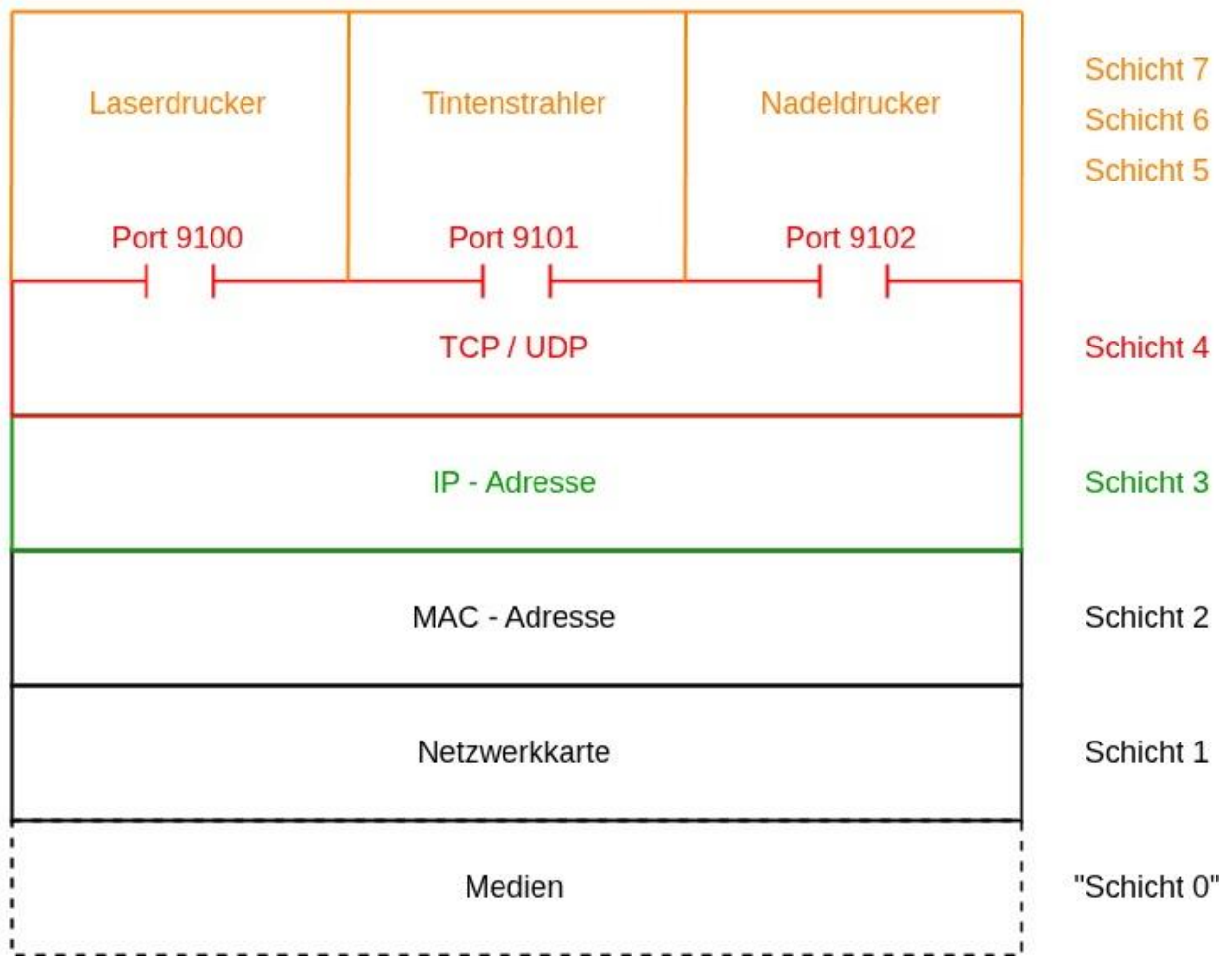
### 9.1.1 Warum Ports?



Betrachten wir diesen uralten Printserver (aus dem Pleistozän der Computertechnik) etwas genauer:

- Die IP-Adresse des Gerätes ist: 10.1.1.1
- An Port 9100 ist ein Laserdrucker angeschlossen.
- An Port 9101 ist ein Tintenstrahler angeschlossen.
- An Port 9102 ist ein Nadeldrucker angeschlossen.
- Ein Druckauftrag wird immer an die IP-Adresse 10.1.1.1 geschickt.
- Der Port entscheidet, auf welchem Drucker der Auftrag aufs Papier gebracht wird.

Drehen Sie im Geiste diesen Printserver nach oben und Sie haben das ganze OSI-Modell in praktischer Darstellung:



siehe /Filius\_Szenen/9.1.1\_ein\_Server\_viele\_Dienste.flv

- Die IP-Adresse (z. B. 192.168.1.11) definiert nur den Host (PC, Server).
- Auf einem PC laufen häufig gleichzeitig mehrere netzwerkfähige Programme, z. B. Mail-Client und Webbrowser.
- Auf einem Server werden oft gleichzeitig mehrere Dienste angeboten, z. B. Webserver, DHCP-Server, File-Server
- Die Dienste müssen unterschieden werden können => verschiedene Ports definieren die verschiedenen Programme oder Dienste
- Anders ausgedrückt: Die postalische Adresse eines Mehrfamilienhauses (IP-Adresse) reicht nicht aus, soll das Paket an Meier oder Lehmann oder Schulze (Ports) ausgeliefert werden?

## 9.1.2 Schreibweisen von IP-Adresse und Port

- IPv4-Adresse:Port

Beispiel: 192.168.1.11:80

- [IPv6-Adresse]:Port

Beispiel: [2001:1234:5678:90AB:CDEF:1A2B:3C4D:5E6F]:80

Die IPv6-Adresse muss(!) in eckige Klammern gesetzt werden.

==>> Die Kombination aus IP-Adresse und Port wird "Socket" genannt.

## 9.1.3 Aufteilung der Ports

### 9.1.3.1 "System Ports"

- früher "well known ports" genannt
- Beginn: Port 0
- Ende: Port 1023
- Ports aus diesem Bereich sind "unantastbar".
- Denken Sie an Nummernschilder von Fahrzeugen wie "Y", "THW", "BP".
- Ports, die man kennen sollte:

Port 20, 21	=> ftp	=> nutzt TCP
Port 22	=> ssh	=> nutzt TCP
Port 25	=> SMTP	=> nutzt TCP
Port 53	=> DNS	=> nutzt meist UDP
Port 67, 68	=> DHCP	=> nutzt immer UDP
Port 80	=> http	=> nutzt TCP
Port 110	=> POP3	=> nutzt TCP
Port 123	=> NTP	=> nutzt meist UDP
Port 143	=> imap	=> nutzt TCP
Port 443	=> https	=> nutzt TCP
Port 445	=> SMB (SaMBa)	=> nutzt TCP

### 9.1.3.2 "User Ports"

- früher "registered ports" genannt
- Beginn: Port 1024
- Ende: Port 49151
- Firmen haben sich aus diesem Bereich Ports für ihre Softwareprodukte "registrieren" lassen.
- Denken Sie an Standard-Nummernschilder von privaten Fahrzeugen.
- Ports, die man eventuell kennt:

Port 3128	=> squid-proxy	=> nutzt TCP
Port 3306	=> MySQL	=> nutzt TCP
Port 9100	=> HP-Druckerport	=> nutzt TCP
Port 10000	=> Webmin	=> nutzt TCP

### 9.1.3.3 "Dynamic/Private Ports"

- hier gab es keine Änderung der Bezeichnung
- Beginn: Port 49152
- Ende: Port 65535
- Diese Ports sind keiner Software fest zugeordnet.
- Diese Ports können nur kurzfristig (temporär) verwendet werden.
- Denken Sie an Nummernschilder für eine Tageszulassung oder an ein Überführungskennzeichen.

## 9.2 Protokolle der Schicht 4

TCP und UDP im Vergleich

TCP	UDP
<ul style="list-style-type: none"><li>* <u>verbindungsorientiert</u><ul style="list-style-type: none"><li>* <u>Verbindungsaufbau</u></li><li>* kontrollierte Datenübertragung</li><li>* <u>Verbindungsabbau</u></li></ul></li><li>* sichere Datenübertragung</li><li>* langsamer</li><li>* ausschließlich 1 : 1 Verbindung</li><li>* fester Algorithmus</li><li>* "Einschreiben mit Rückschein"</li></ul>	<ul style="list-style-type: none"><li>* <u>verbindungslos</u><ul style="list-style-type: none"><li>* -----</li><li>* -----</li><li>* -----</li></ul></li><li>* <u>unsichere</u> Datenübertragung (Datenverlust ?)</li><li>* schneller</li><li>* 1 : 1 und 1 : n - Verbindung möglich (Broadcast)</li><li>* "Spielwiese" für Programmierer</li><li>* "Postkarte"</li></ul>

### 9.2.1 UDP

Vermutlich DAS Protokoll der Zukunft.

siehe QUIC

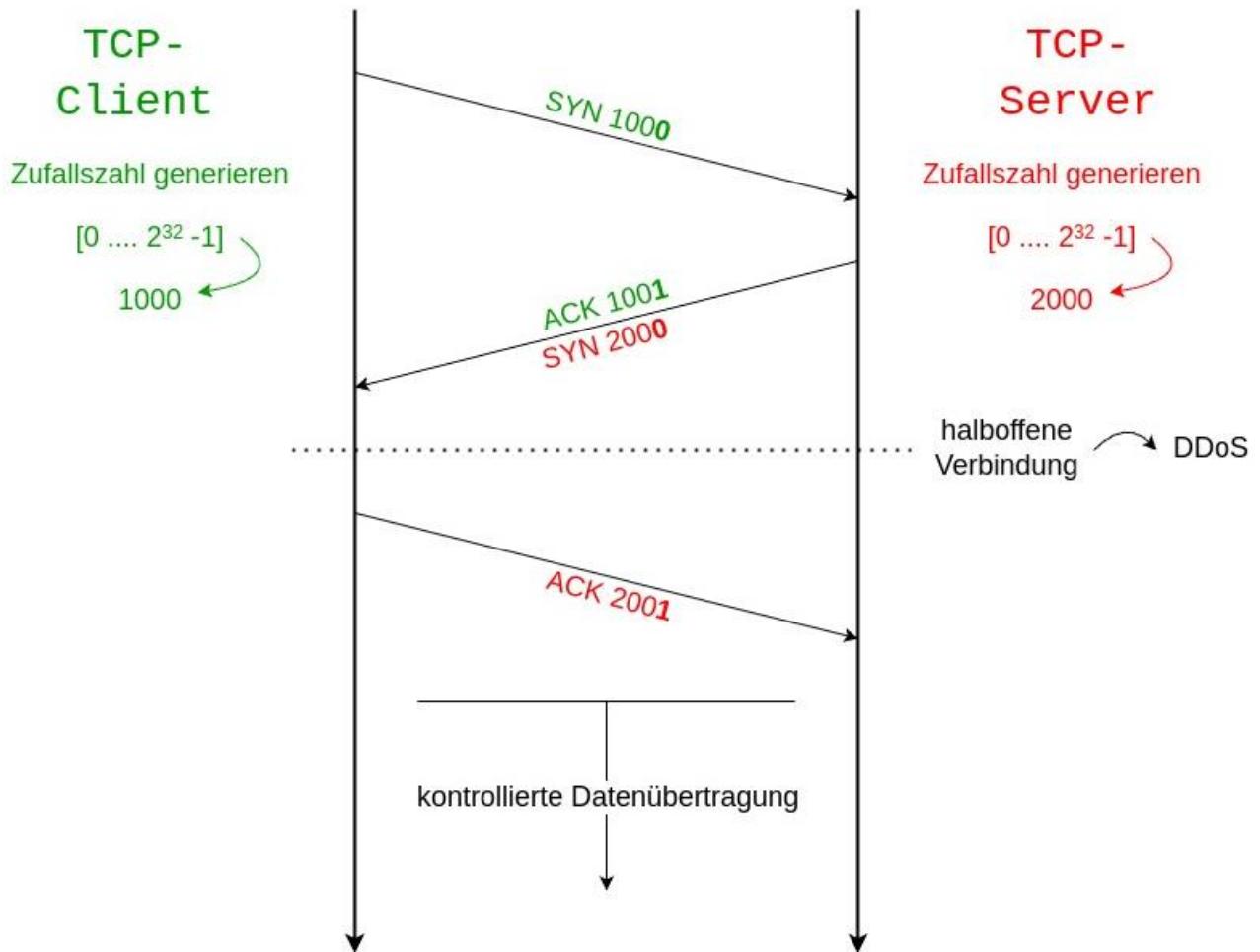
### 9.2.2 TCP

[de.wikipedia.org/wiki/Transmission Control Protocol](https://de.wikipedia.org/wiki/Transmission_Control_Protocol)

hat viele Schwachstellen

aufwendiger Verbindungsaufbau bei verschlüsselter Verbindung

### 9.2.2.1 Verbindungsaufbau zwischen Client und Server



- Client sendet an den Server seine Zufallszahl (z. B. 1000) und setzt das SYN-Flag (ich möchte mich mit dir synchronisieren).
- Server sendet die Zufallszahl des Clients um 1 erhöht (1001) zurück und setzt das ACK-Flag. Im gleichen Datenpaket sendet der Server seine Zufallszahl (z. B. 2000) und setzt das SYN-Flag (auch ich möchte mich mit dir synchronisieren).
- Client sendet die Zufallszahl des Servers um 1 erhöht (2001) zurück und setzt das ACK-Flag.
- Die Verbindung wurde AUFgebaut.
- 3 "Pfeile" => 3 Datenpakete sind für den Verbindungsaufbau nötig => 3-Wege-Handshake-Verfahren
- ACHTUNG: DDoS-Angriffe möglich!

[de.wikipedia.org/wiki/Denial\\_of\\_Service](https://de.wikipedia.org/wiki/Denial_of_Service)  
[de.wikipedia.org/wiki/SYN-Flood](https://de.wikipedia.org/wiki/SYN-Flood)

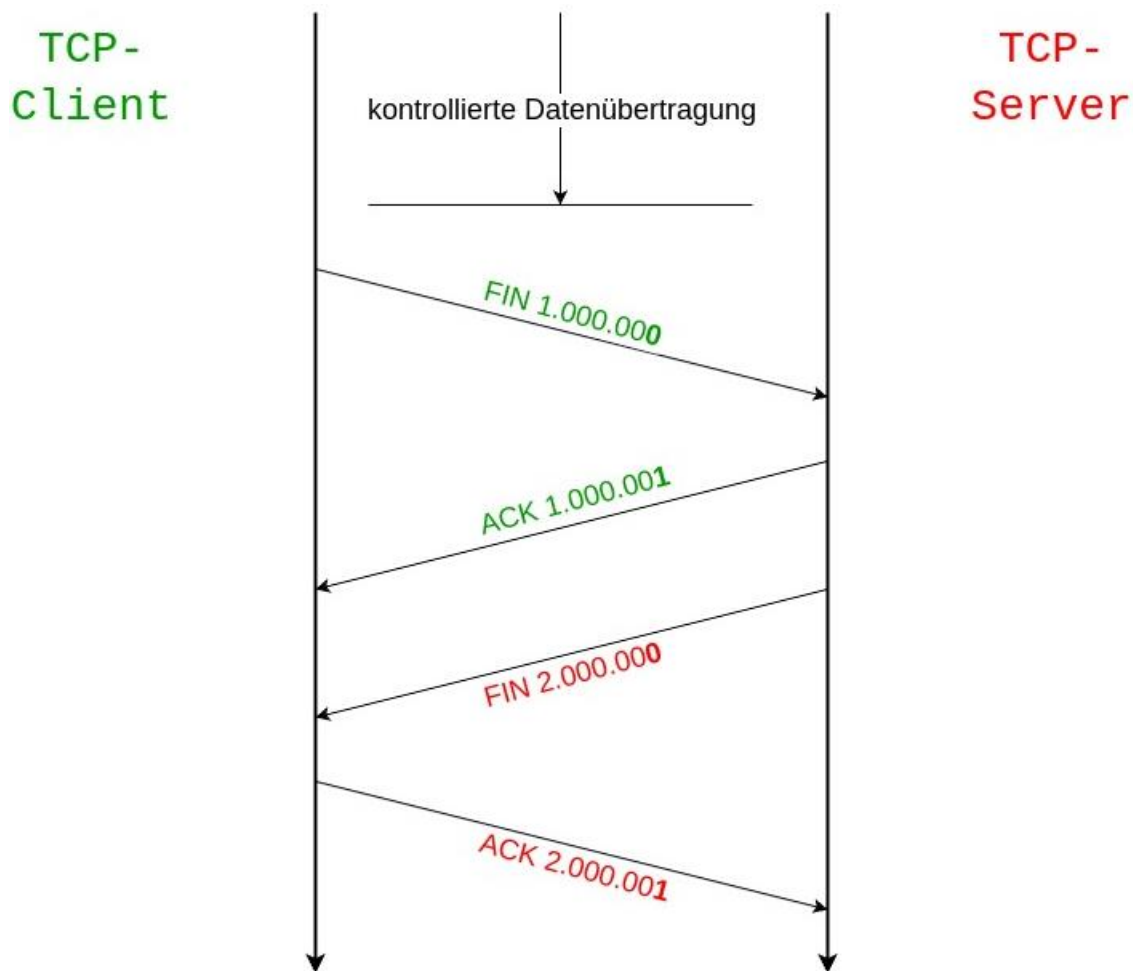
### 9.2.2.2 Die kontrollierte Datenübertragung bei TCP

[youtube.com/watch?v=c5qqPo5v3-U](https://www.youtube.com/watch?v=c5qqPo5v3-U)

Eine bessere Erklärung ist kaum zu finden.

siehe /Filius\_Szenen/9.2.2.2\_WebSrv\_viele\_Daten.flv

### 9.2.2.3 VerbindungsABbau zwischen Client und Server



- Client sendet an den Server ein FIN-Flag (ich möchte die Verbindung beenden).
- Server sendet an den Client ein ACK-Flag zurück (ok, einverstanden).
- Server sendet an den Client auch ein FIN-Flag (auch ich möchte die Verbindung beenden).
- Client sendet an den Server auch ein ACK-Flag (ok, auch einverstanden).
- Die Verbindung wurde ABgebaut.
- 4 "Pfeile" => 4 Datenpakete sind für den VerbindungsABbau nötig => 4-Wege-Handshake-Verfahren

### 9.2.2.4 TCP-Header

siehe Westermann, Seite 581

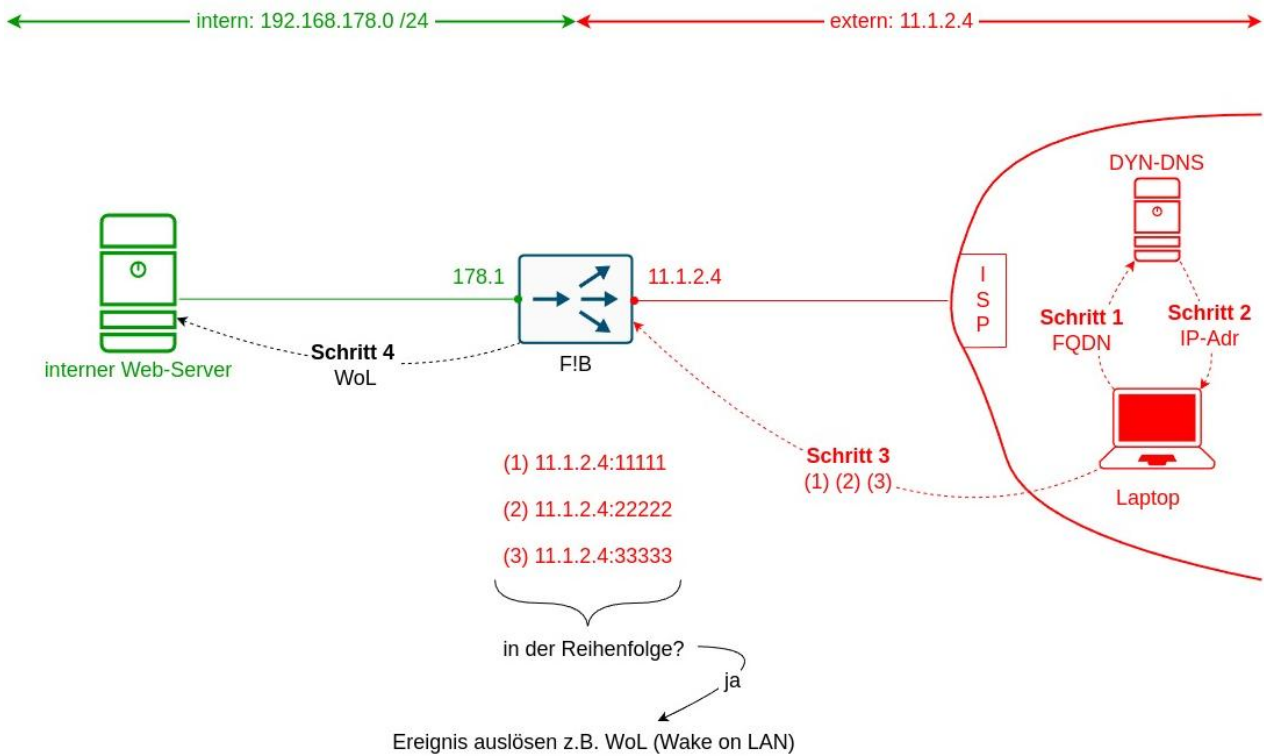
[de.wikipedia.org/wiki/Transmission\\_Control\\_Protocol#Aufbau des TCP-Headers](https://de.wikipedia.org/wiki/Transmission_Control_Protocol#Aufbau_des_TCP-Headers)

### 9.2.3 QUIC

[de.wikipedia.org/wiki/QUIC](https://de.wikipedia.org/wiki/QUIC)

- soll zukünftig die Vorteile von UDP und TCP vereinen.
- arbeitet immer mit der Verschlüsselung der Daten.

## 9.3 Portknocking



### 9.3.1 Zielsetzung

"Richtiges Anklopfen" aus dem externen Netz => löst ein Ereignis im internen Netz aus.

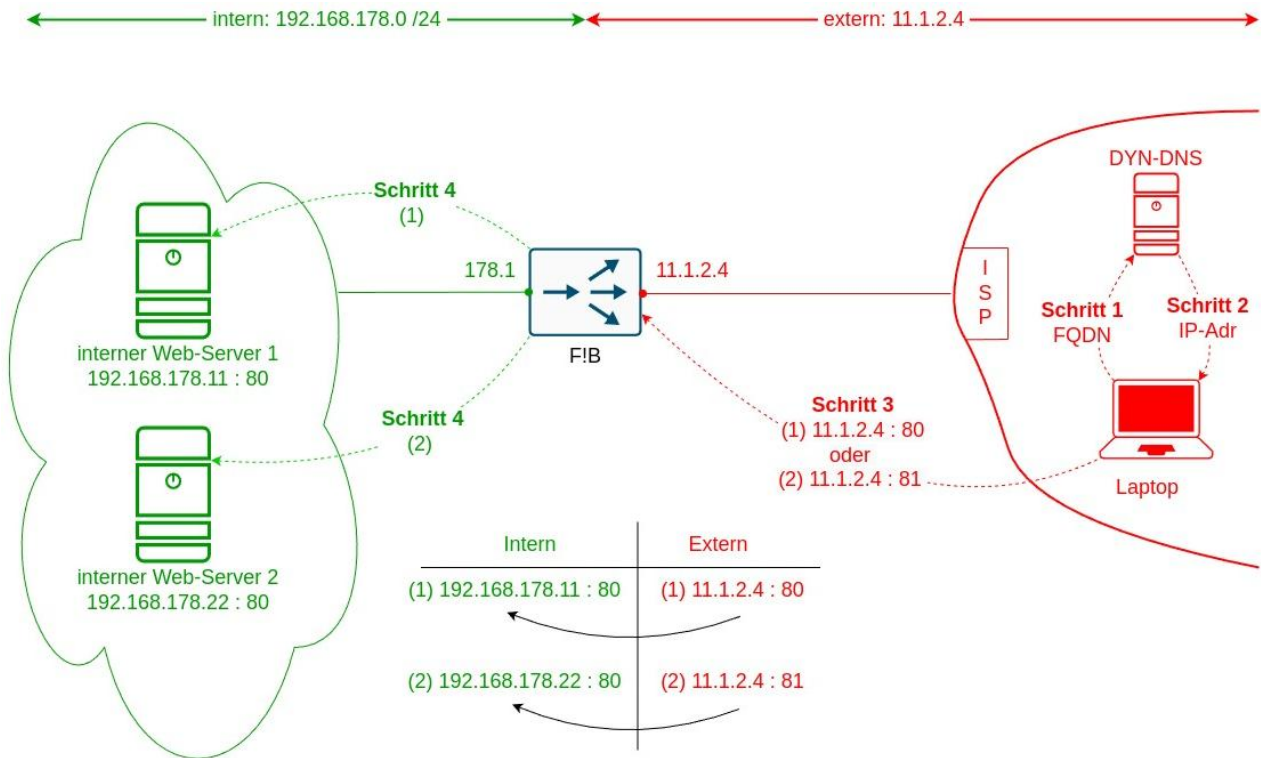
### 9.3.2 Beschreibung

- Die "Fritz!Box" (FB) hat vom ISP eine externe IP-Adresse (11.1.2.4) erhalten, über die sie auch aus dem Internet erreichbar ist.
  - Auf der Fritz!Box wurde Portknocking aktiviert.
  - Auf der Fritz!Box wurde eine Tabelle mit 3 Sockets erstellt:
    - 1.) externe IP-Adresse und erster willkürlicher Port (11.1.2.4:1111)
    - 2.) externe IP-Adresse und zweiter willkürlicher Port (11.1.2.4:2222)
    - 3.) externe IP-Adresse und dritter willkürlicher Port (11.1.2.4:3333)
  - Auf der Fritz!Box wurde ein Ereignis definiert, das ausgelöst wird, wenn die 3 Sockets in der richtigen Reihenfolge 1.), 2.), 3.) angesprochen werden (im Beispiel Wake on LAN).  
[de.wikipedia.org/wiki/Wake\\_On\\_LAN](http://de.wikipedia.org/wiki/Wake_On_LAN)
  - Sendet das Laptop aus dem Internet 3 Datenpakete in der richtigen Reihenfolge:
    - (1) 11.1.2.4:1111
    - (2) 11.1.2.4:2222
    - (3) 11.1.2.4:3333
- => WoL wird ausgelöst, der interne Webserver (192.168.178.11) bootet.

## 9.4 Portforwarding / Destination NAT

[de.wikipedia.org/wiki/Netzwerkadressübersetzung](https://de.wikipedia.org/wiki/Netzwerkadressübersetzung)

siehe /Filius\_Szenen/9.4\_Port\_forwarding.flv



### 9.4.1 Zielsetzung

Weiterleitung von Anfragen aus dem externen Netz (Internet) in das interne, private Netz

### 9.4.2 Begriffe

Aus der Unix-Welt stammt der Begriff "Destination-NAT" (sinngemäß: Übersetzung am Ziel).

### 9.4.3 Beschreibung

- Die "Fritz!Box" (FB) hat vom ISP eine externe IP-Adresse (11.1.2.4) erhalten, über die sie aus dem Internet erreichbar ist.
- Aus dem Internet soll ein Laptop auf interne Geräte zugreifen können.
- Im internen Netz existieren 2 Web-Server, die von extern erreichbar sein sollen:

Web-Server 1:

- IP-Adresse: 192.168.178.11
- Port: 80

Web-Server 2:

- IP-Adresse: 192.168.178.22
- Port: 80

- Auf der Fritz!Box wurde Portforwarding aktiviert.
- Auf der Fritz!Box wurden 2 willkürliche externe Ports gewählt (80 und 81).
- Auf der Fritz!Box wurde eine Tabelle mit 2 Einträgen erstellt:

(1) 11.1.2.4:80 wird auf 192.168.178.11:80 weitergeleitet

(2) 11.1.2.4:81 wird auf 192.168.178.22:80 weitergeleitet

==>> Die externe IP-Adresse muss immer gleich sein, nur die willkürlichen externen Ports dienen der Unterscheidung, welches interne Gerät angesprochen werden soll.

- Greift das Laptop mit (1) von extern auf die externe IP-Adresse 11.1.2.4 und den willkürlichen externen Port 80 zu, wird die Anfrage nach intern auf die IP-Adresse 192.168.178.21 und den Port 80 (Web-Server 1) weitergeleitet.
- Greift das Laptop mit (2) von extern auf die externe IP-Adresse 11.1.2.4 und den willkürlichen externen Port 81 zu, wird die Anfrage nach intern auf die IP-Adresse 192.168.178.22 und den Port 80 (Web-Server 2) weitergeleitet.

### 9.4.4 Anmerkungen

==>> Portforwarding ist nicht sehr sicher:

- Von extern wird ein Zugang zum internen Netz geschaffen => kann zum Problem werden.
- Portforwarding kann mit Portknocking und WoL (siehe oben) kombiniert werden

Portknocking => Ereignis auslösen (WoL)

WoL => interner Webserver bootet

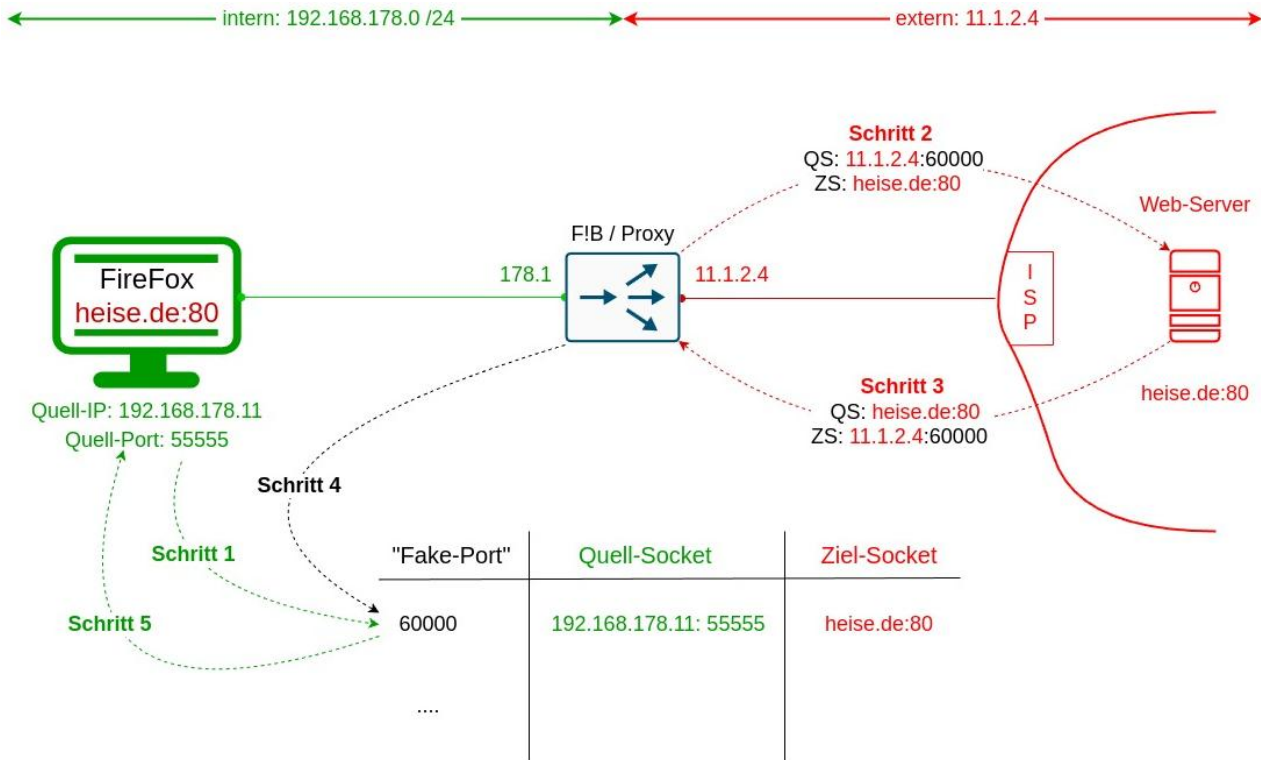
Portforwarding => Zugriff auf den internen Webserver

## 9.5 NAT (PAT) / Source NAT

[de.wikipedia.org/wiki/Port\\_Address\\_Translation](https://de.wikipedia.org/wiki/Port_Address_Translation)

siehe nochmals /Filius\_Szenen/8.5.2\_Home\_Berlin\_Paris\_DNS\_einfach.flv

=> den Datenaustausch der Fritz!Box an beiden Ports betrachten!



### 9.5.1 Zielsetzung

Weiterleitung von Anfragen aus dem internen, privaten Netz in das externe Netz (Internet)

### 9.5.2 Begriffe

==>> Dieser Mechanismus heißt eigentlich PAT, aber umgangssprachlich wird von NAT gesprochen.

- Leider ist dieses Problem auch schon in den IHK-Prüfungen aufgetaucht!
- Aus der Unix-Welt stammt der Begriff "Source-NAT" (sinngemäß: Übersetzung an der Quelle).

### 9.5.3 Beschreibung

- Die "Fritz!Box" (F!B) hat vom ISP eine externe IP-Adresse (11.1.2.4) erhalten, mit der sie ins Internet "gehen" kann.
- Die F!B hat eine dynamische Tabelle mit 3 Spalten:
  1. Spalte "Fake-Port": Ports aus dem Bereich der "Dynamic/Private Ports" (siehe oben)
  2. Spalte "Quell-Socket": für die Anfragen von den internen Hosts, hier im Beispiel für den PC.
  3. Spalte "Ziel-Socket": für die Anfragen, welche Webseiten von der F!B für die internen Hosts geholt werden sollen.
- Intern existiert ein privater IP-Adressbereich „Scope“ (siehe private IP-Adressen), durch ihre private IP-Adresse haben alle internen Hosts keine Möglichkeit, direkt ins Internet "zu gehen".
- Der PC hat einen Browser geöffnet und möchte die Webseite von heise.de (193.99.144.85:80) holen.
- Durch das Öffnen des Browsers entsteht beim PC ein Socket: 192.168.178.11:55555.

### 9.5.4 Ablauf (step by step) für den PC

- Schritt 1:

Der PC (192.168.178.11:55555) schickt seinen Wunsch nach der Webseite heise.de (193.99.144.85:80) an die F!B.

Die F!B trägt diesen Wunsch in der Zeile des "Fake-Ports" 60000 ein.

In dieser Zeile stehen jetzt der Quell-Socket: 192.168.178.11:55555 und der Ziel-Socket: 193.99.144.85:80 (heise.de).
- Schritt 2:

Die F!B fordert die Webseite von heise.de an, dazu nutzt sie ihre externe IP-Adresse und den Fake-Port 60000 als Quell-Socket.

Inhalt der Anfrage der Fritz!Box (sinngemäß):

Quell-Socket: 11.1.2.4:60000 (externe IP-Adresse der F!B und Fake-Port)

Ziel-Socket : 193.99.144.85:80 (heise.de und Webserver-Port)

"Gib mir die Webseite."
- Schritt 3:

Antwort von heise.de (sinngemäß):

Quell-Socket: 193.99.144.85:80 (heise.de und Webserver-Port)

Ziel-Socket : 11.1.2.4:60000 (externe IP-Adresse der F!B und Fake-Port)

"Hier kommt der Inhalt der Webseite."
- Schritt 4:

Nach der Lieferung der Webseite von heise.de:80 schaut die F!B unter dem Fake-Port 60000 nach, von wem die Bestellung ursprünglich kam:

=> 192.168.178.11:55555.
- Schritt 5:

Die F!B liefert abschließend die Webseite von heise.de:80 an den PC 192.168.178.11 mit seinem geöffneten Port 55555.

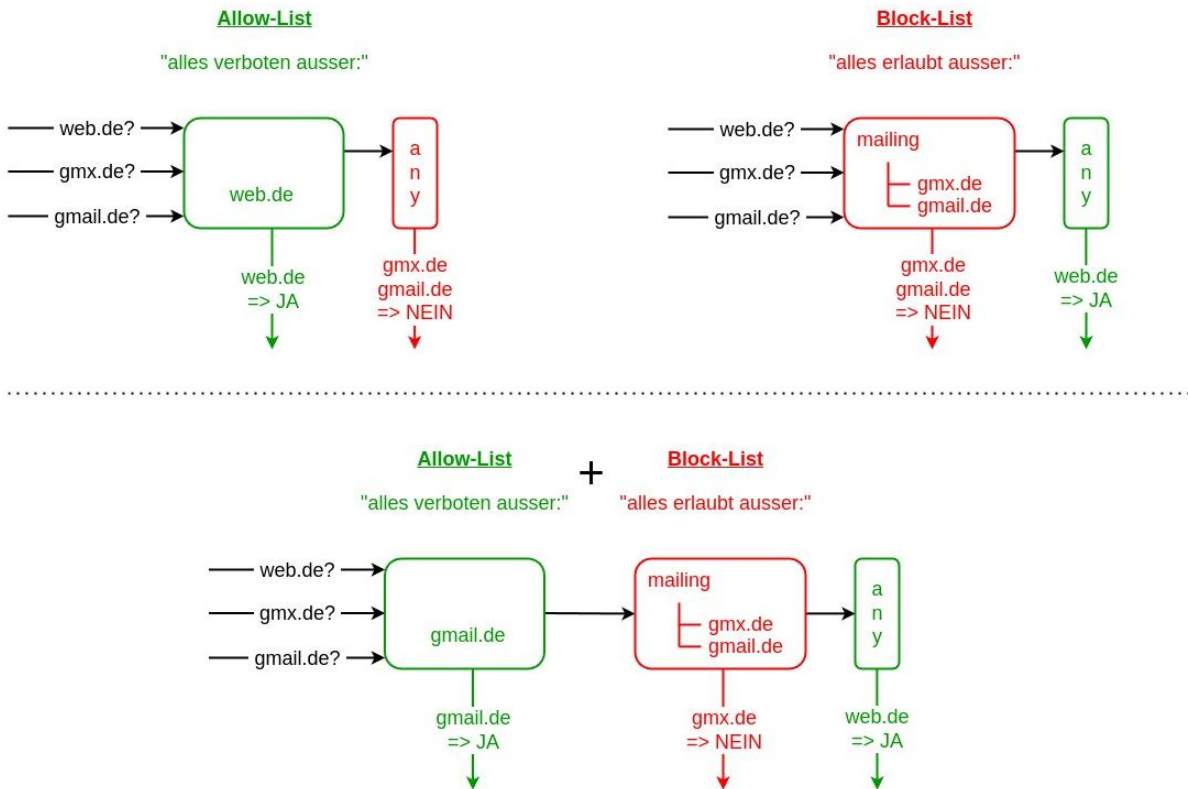
## 9.6 Black- und Whitelist

==>> heute Block- und Allowlist

(rot => Blocklist, grün => Allowlist)

[dsi.ut-capitole.fr/blacklists/download/](http://dsi.ut-capitole.fr/blacklists/download/)

siehe /Filius\_Szenen/9.6\_SPI\_BL\_berlin\_paris\_nfe.fl



### 9.6.1 Allowlist

=> "Alles verboten außer . . . ."

- Eine Allowlist ist eine „ewige“ Baustelle.
- Täglich kommen neue Wünsche der Mitarbeiter.
- Ist ein sehr striktes Mittel, um das Firmennetz vor Missbrauch zu schützen.

Allowlist (Auszug):

„web.de“

- Anfrage an die Allowlist: web.de erlaubt? => steht in der Allowlist  
=> Antwort JA
- Anfrage an die Allowlist: gmx.de erlaubt? => steht NICHT in der Allowlist  
=> Antwort NEIN
- Anfrage an die Allowlist: gmail.de erlaubt? => steht NICHT in der Allowlist  
=> Antwort NEIN  
=> Alle Mitarbeiter dürften ausschließlich auf die Webseite von web.de zugreifen.

### 9.6.2 Blocklist

=> „Alles erlaubt außer . . . .“

- Blocklist wird meist von einem externen Anbieter bezogen (darf laut Lizenz oft NICHT editiert werden) . . . . . und ist trotzdem niemals vollständig.
- Stündlich kommen neue "schwarze" Seiten ins Internet.
- Ist aber ein recht brauchbares Mittel, um das Firmennetz vor Missbrauch zu schützen.

Blocklist (Auszug):

„gmx.de“

„gmail.de“

- Anfrage an die Blocklist: web.de erlaubt? => steht NICHT in der Blocklist  
=> Antwort JA
- Anfrage an die Blocklist: gmx.de erlaubt? => steht in der Blocklist  
=> Antwort NEIN
- Anfrage an die Blocklist: gmail.de erlaubt? => steht in der Blocklist  
=> Antwort NEIN  
=> Mitarbeiter dürfen nur auf web.de zugreifen.
- Weil die Blocklist „zu streng“ ist, darf auch niemand zu gmail.de.  
=> keine optimale Lösung

### 9.6.3 Kombination aus Allowlist UND Blocklist

- Nutze eine sehr umfangreiche Blocklist und "bohre" sie mithilfe der Allowlist an den Stellen auf, an denen sie "zu streng" ist.
- Dabei wird die Allowlist zuerst durchlaufen und dann erst die Blocklist.

Allowlist (Auszug):

„gmail.de“

Blocklist (Auszug):

„gmx.de“

„gmail.de“

- Anfrage an Block- UND Allowlist: web.de erlaubt? => steht in keiner Liste  
=> Antwort JA
- Anfrage an Block- UND Allowlist: gmx.de erlaubt? => steht in der Blocklist  
=> Antwort NEIN
- Anfrage an Block- UND Allowlist: gmail.de erlaubt? => steht in der Allowlist  
=> Antwort JA  
=> Alle Mitarbeiter können auf web.de und gmail.de => praktikable Lösung
- Häufig sind Blocklists in Kategorien eingeteilt:  
Dating  
Mailing  
Hacking  
.....

## 9.6.4 Squidguard

- Mit Squidguard können ganze Kategorien ein- und ausgeschaltet werden.
- Die logische Syntax ist einfach: ein „!“ => bedeutet ein NICHT, wie in vielen Programmiersprachen:  
„Allowlist !Dating !Mailing !Hacking.....any“
- durchlaufe Allowlist
  - bei Übereinstimmung => ERLAUBT                      => Anfrage beendet
  - KEINE Übereinstimmung gefunden?                      => gehe zu !Dating
- durchlaufe !Dating
  - bei Übereinstimmung => VERBOTEN    => Anfrage beendet
  - KEINE Übereinstimmung gefunden?    => gehe zu !Mailing
- durchlaufe !Mailing
  - usw.
  - bei Übereinstimmung => VERBOTEN    => Anfrage beendet
  - KEINE Übereinstimmung gefunden?    => gehe zu => any => ERLAUBT => Anfrage beendet.

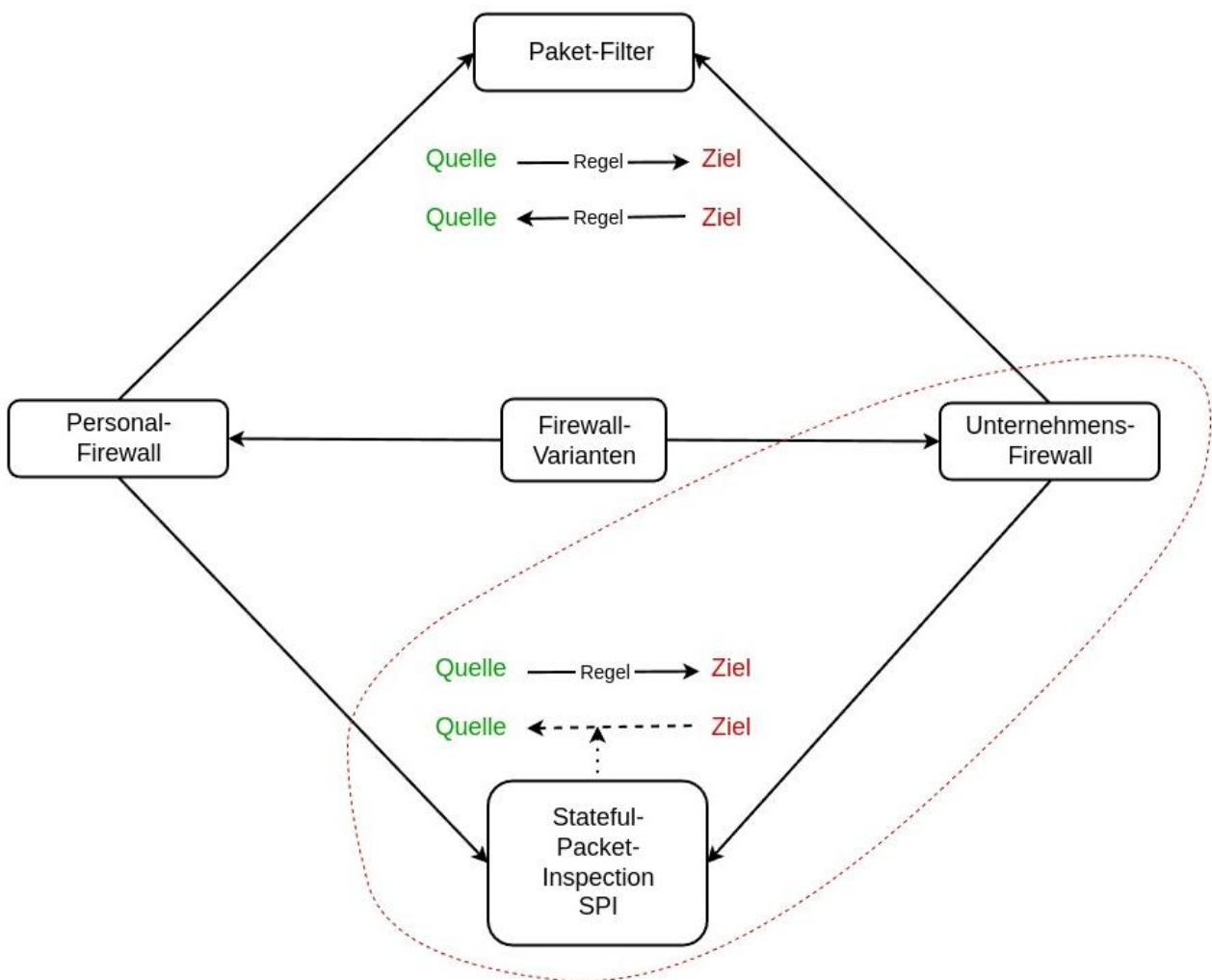
## 10 Firewalls

- (im weiteren Verlauf oft mit "FW" abgekürzt)

[de.wikipedia.org/wiki/Firewall](https://de.wikipedia.org/wiki/Firewall)

[de.wikipedia.org/wiki/Stateful\\_Packet\\_Inspection](https://de.wikipedia.org/wiki/Stateful_Packet_Inspection)

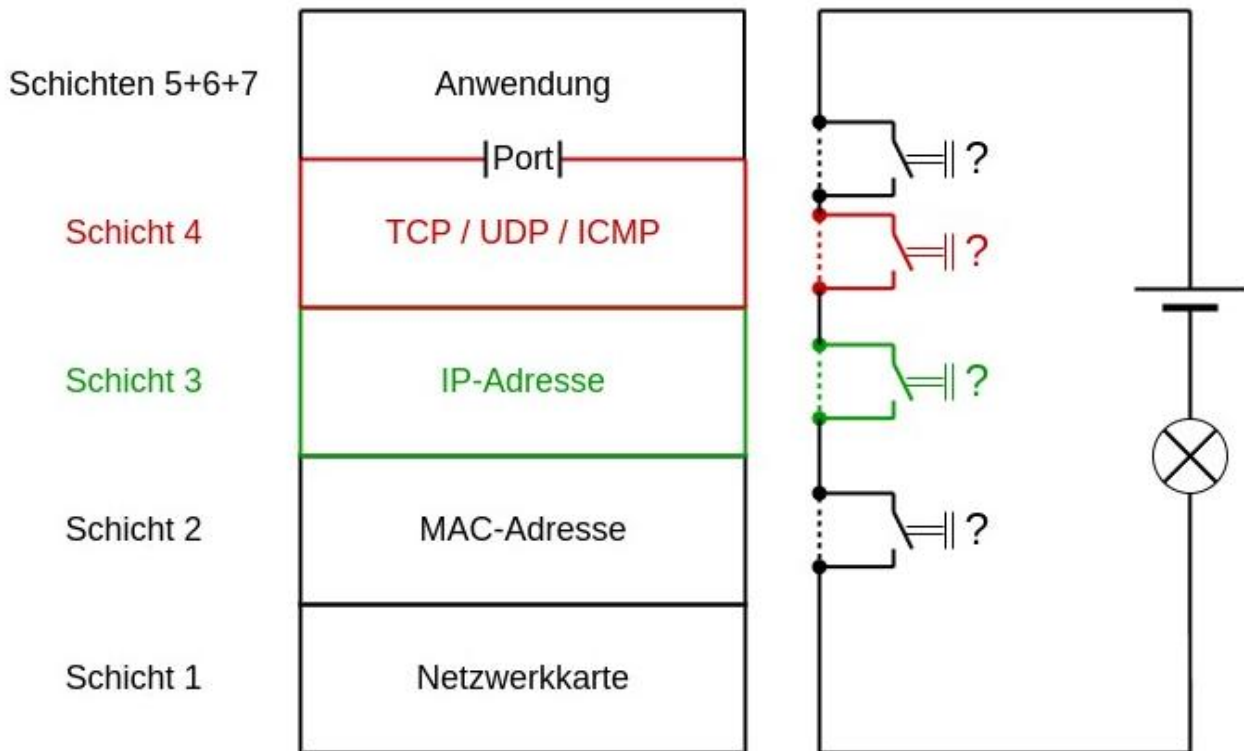
siehe /Filius\_Szenen/10\_BFW\_SPI\_BL\_berlin\_paris\_nfe.fls



## 10.1 Unterscheidungsmerkmale

- Personal-FW => schützt nur den Host (PC, Server), auf dem sie läuft
- Unternehmens-FW => Schützt das LAN oder Teile des LANs eines Unternehmens
- Paketfilter-FW:
  - kann als Personal-FW oder auch als Unternehmens-FW eingesetzt werden
  - ist veraltet und unsicher
  - Sowohl Hinweg als auch Rückweg müssen angegeben werden:
    - Quelle => Ziel
    - Quelle <= Ziel
  - ==>> Noch immer Thema in den IHK-Prüfungen!
- SPI-FW:
  - Kann als Personal-FW oder auch als Unternehmens-FW eingesetzt werden.
  - Ist modern und recht sicher.
  - Nur der Hin-Weg muss angegeben werden:
    - Quelle => Ziel
  - Der Rückweg (Quelle <= Ziel) wird durch die SPI-FW selbst gesteuert.
  - Mit "iptables" ist eine kostenlose und leistungsstarke Firewall-Lösung unter Linux verfügbar.
  - ==>> Alle weiteren Ausführungen beziehen sich auf eine SPI-FW mit iptables im Rahmen einer Unternehmens-Firewall, siehe rot gestrichelte Linie in der Grafik.

## 10.2 Steuerbare OSI-Schichten der SPI-FW unter Linux



=> Alle Bedingungen sind logisch UND-verknüpft!

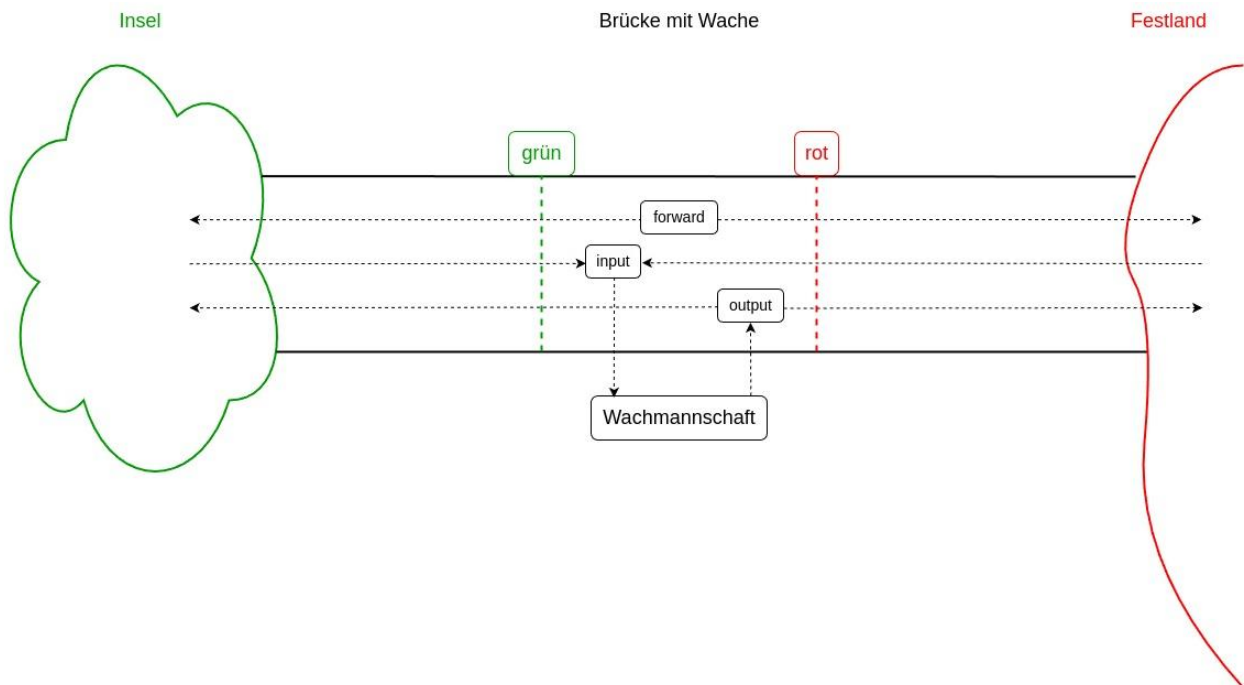
- Schicht 2 => MAC-Adresse: Nur der Rechner mit der entsprechenden MAC-Adresse darf....  
UND
- Schicht 3 => IP-Adresse: Nur der Rechner mit der entsprechenden IP-Adresse darf....  
UND
- Schicht 4 => TCP oder UDP: Nur wenn der Rechner ein Datenpaket über TCP oder UDP sendet, darf er....  
UND
- Schichten 5 - 7 => Port: Nur wenn der Rechner über Port xy kommuniziert, darf er....
- Port == Anwendung

==>> Dabei gilt: Was NICHT abgefragt wird, ist erlaubt!

## 10.3 Firewall als Brücke zwischen einer Insel und dem Festland

Nehmen wir die Insel Sylt als unser zu sicherndes LAN (grünes Netz):

[de.wikipedia.org/wiki/Sylt](https://de.wikipedia.org/wiki/Sylt)



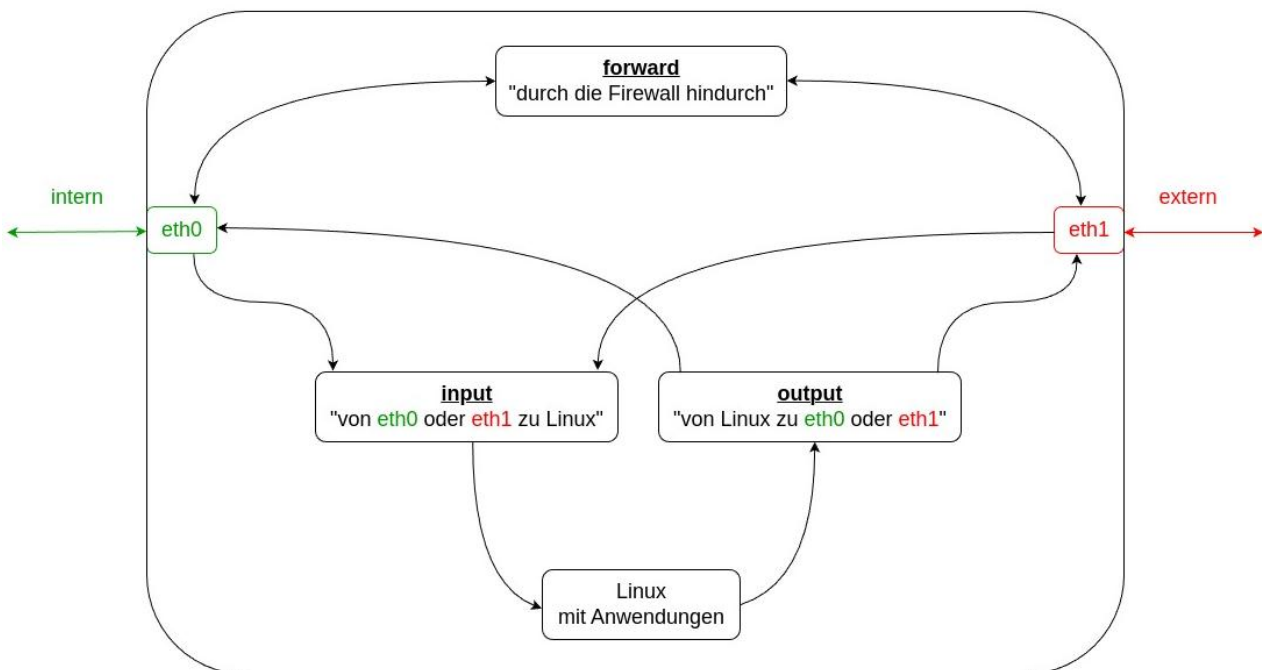
Zwischen der Insel Sylt und dem Festland gibt es nur eine Verbindung, über den Hindenburgdamm.

Folgende Denke:

- Auf dem Hindenburgdamm gibt es eine Brücke mit einer Wachmannschaft.
- Die Brücke hat 2 Schranken.
- GRÜNE Schranke:  
In Richtung Insel Sylt (zum sicheren "internen Netz").
- ROTE Schranke:  
In Richtung Festland (zum unsicheren "externen Netz").
- Die Wachmannschaft hat 3 Zettel (Regelsätze genannt), auf denen bestimmte Regeln stehen:
  1. Zettel „FORWARD“:  
Wer darf vom Festland auf die Insel?  
Wer darf von der Insel auf das Festland?
  2. Zettel „INPUT“:  
Wer darf, vom Festland kommend, die Wachmannschaft besuchen?  
Wer darf, von der Insel kommend, die Wachmannschaft besuchen?
  3. Zettel „OUTPUT“:  
Wer von der Wachmannschaft darf in Richtung Festland gehen?  
Wer von der Wachmannschaft darf in Richtung Insel gehen?

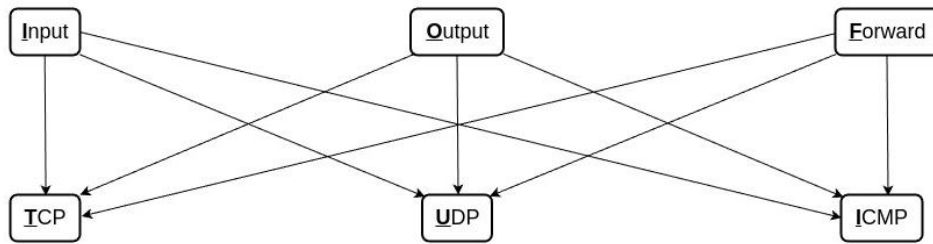
## 10.4 Innerer Aufbau der SPI-FW mit iptables

Firewall innen vereinfacht



- Wachmannschaft => Betriebssystem Linux (mit installierten Anwendungen)
- GRÜNE Schranke => Netzwerkkarte eth0 (in diesem Beispiel)
- ROTE Schranke => Netzwerkkarte eth1 (in diesem Beispiel)

=> Bei iptables können wir die 3 Regelsätze (Input, Output, Forward) mit den 3 Protokollen (TCP, UDP, ICMP) kombinieren und daraus eine Tabelle erarbeiten.




---

Regel	Input Output Forward	TCP UDP ICMP	Quell-IP	Quell-MAC	Ziel-IP	Ziel-Port(s)	Kommentar
(last)							Alles andere wird verboten!

- FORWARD:

eth0 => Firewall => eth1

eth1 => Firewall => eth0

Hauptaufgabe von FORWARD:

Wer darf welche Daten von eth0 => eth1 senden?

Wer darf welche Daten von eth1 => eth0 senden?

- INPUT:

eth0 => Betriebssystem

eth1 => Betriebssystem

Hauptaufgabe von INPUT:

Wer darf welche Daten über eth0 => Betriebssystem senden?

Wer darf welche Daten über eth1 => Betriebssystem senden?

- OUTPUT:

Betriebssystem Linux => eth0

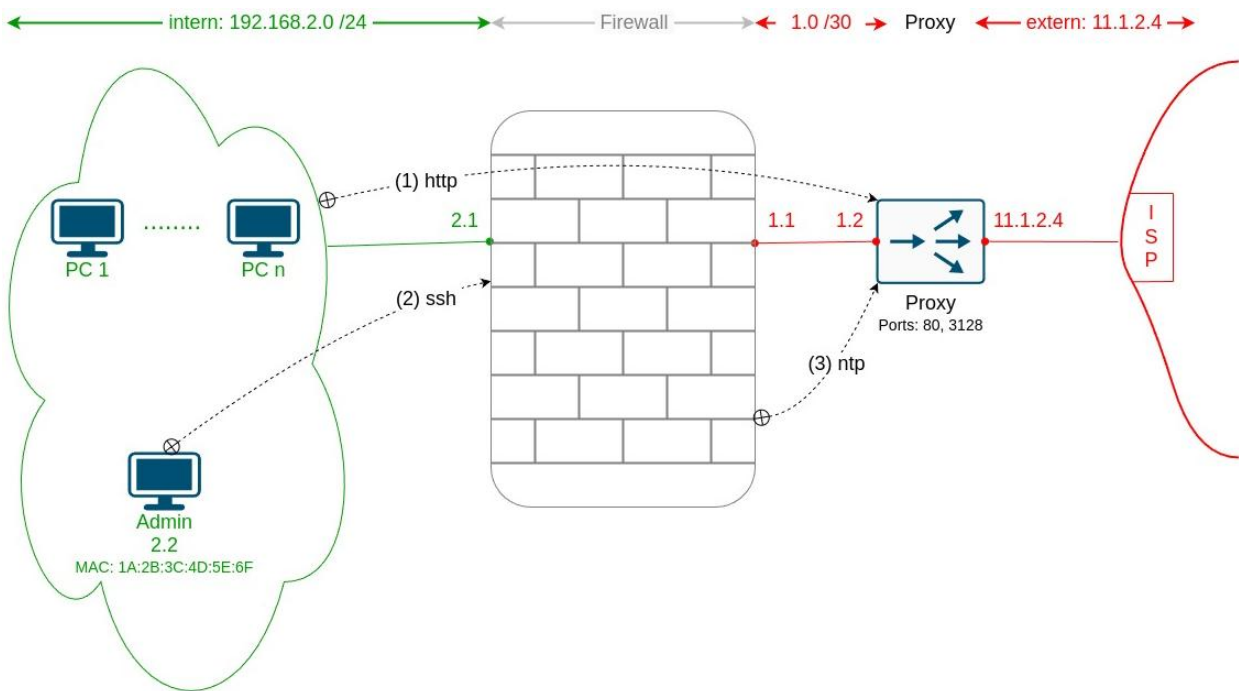
Betriebssystem Linux => eth1

Hauptaufgabe von OUTPUT:

Welche Daten darf das Betriebssystem über eth0 aussenden?

Welche Daten darf das Betriebssystem über eth1 aussenden?

## 10.5 Beispiel einer einfachen Firewall für den praktischen Einstieg



- jeder Pfeil hat einen Anfang (Kreis mit x gefüllt) => Quelle
- jeder Pfeil hat ein Ende (Spitze) => Ziel
- Der Rückweg ist hier uninteressant, wir haben schließlich eine SPI-Firewall!
- Überlegen Sie für jeden Pfeil:  
Ist es FORWARD oder INPUT oder OUTPUT?
- Ergibt es Sinn, die MAC-Adresse in die Regel mit aufzunehmen?
- ist die IP-Adresse ein ganzes Netz => Angabe der Subnetzmaske (hier: /24)
- ist die IP-Adresse ein einzelner Host => keine Angabe der Subnetzmaske
- Wird TCP oder UDP oder ICMP genutzt?
- Welcher Port wird genutzt? (ICMP hat keinen Port)

Füllt man die Tabelle entsprechend der Skizze aus, ergibt sich folgendes Bild:

Regel	Input Output	TCP UDP ICMP	Quell-IP	Quell-MAC	Ziel-IP	Ziel-Port(s)	Kommentar
(1)	Forward	TCP	192.168.2.0/24		192.168.1.2	80,3128	LAN => Proxy
(2)	Input	TCP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.2.1	22	Admin => Firewall
(3)	Output	UDP	192.168.1.1		192.168.1.2	123	Firewall => Proxy
<b>(last)</b>	<b>Alles andere wird verboten!</b>						

Nutzt man nun die Datei /lptables\_Web/iptables.html und überträgt die Tabelle, erhält man folgende Lösung:

	Direction	Protocol	Source IP	Source MAC	Target IP	Port(s)	Comment
<input type="checkbox"/>	FORWARD	TCP	192.168.2.0/24		192.168.1.2	80,3128	Regel (1)
<input type="checkbox"/>	INPUT	TCP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.2.1	22	Regel (2)
<input type="checkbox"/>	OUTPUT	UDP	192.168.1.1		192.168.1.2	123	Regel (3)

Unterhalb der Tabelle finden Sie die Syntax für Iptables.

### 10.5.1 Iptables-Syntax betrachten

Ein bisschen „reverse engineering“ hat noch nie geschadet,  
hier ein Ausschnitt aus der entstandenen Datei „alle.sh“:

```
# start of user-defined rules

echo "Regel (1)"
$FT $MP -s 192.168.2.0/24 -d 192.168.1.2 --dports 80,3128 $R

echo "Regel (2)"
$IT $MAC 1A:2B:3C:4D:5E:6F -s 192.168.2.2 -d 192.168.2.1 --dport 22 $R

echo "Regel (3)"
$OU -s 192.168.1.1 -d 192.168.1.2 --dport 123 $R

# end of user-defined rules

echo "catch all"
$IPT -A INPUT -j DROP
$IPT -A OUTPUT -j DROP
$IPT -A FORWARD -j DROP
```

Erklärungen und Vereinfachungen:

```
echo "Regel (1)"  => Ausgabe der Meldung beim Start des Skripts
"$FT"           => FORWARD TCP
"$MP"           => multiport (Angabe mehrerer Ports auf einmal)
„-s“           => Quell-IP/Subnetzmaske
„-d“           => Ziel-IP
„--dports“     => Ziel-Ports (Achtung Plural!)
„$R“           => das Ende jeder Zeile
                => SPI-FW => ACCEPT (Regel ist erlaubt)

echo "Regel (2)" => Ausgabe der Meldung beim Start des Skripts
"$IT"           => INPUT TCP
"$MAC"         => mac-source (überprüfe auch die MAC-Adresse)
„-s“           => Quell-IP/Subnetzmaske
„-d“           => Ziel-IP
„--dport“     => Ziel-Port (Achtung Singular!)

„$R“ => das Ende jeder Zeile => SPI-FW => ACCEPT (Regel ist erlaubt)

echo "Regel (3)" => Ausgabe der Meldung beim Start des Skripts
"$OU"          => OUTPUT UDP
```

„-s“                   => Quell-IP/Subnetzmaske  
„-d“                   => Ziel-IP  
„--dport“           => Ziel-Port (Achtung Singular!)  
„\$R“                   => das Ende jeder Zeile => SPI-FW => ACCEPT (Regel ist erlaubt)

INPUT / OUTPUT / FORWARD drop => verwirf alle Datenpakete, die nicht den oben aufgeführten Regeln entsprechen

=> Alles andere wird verboten!

## **10.5.2 Umsetzung in der Praxis**

Wie kann man nun dieses Skript nutzen, was wird benötigt?

- ein PC mit Betriebssystem Linux (getestet mit Debian und Ubuntu)
- PC braucht 2 Netzwerkkarten:
  1. Netzwerkkarte => IP-Adresse: 192.168.1.1
  2. Netzwerkkarte => IP-Adresse: 192.168.2.1
- zum "root" auf dem PC werden => "sudo -s" und das Passwort eingeben
- Skript auf den PC kopieren (auch per USB-Stick möglich)
- Skript "ausführbar" machen => "chmod 755 alle.sh"
- Skript starten => "./alle.sh"
- Skript wird abgearbeitet => die Zeilen mit "echo" werden angezeigt

## 10.6 Demilitarisierte Zone ("DMZ")

[de.wikipedia.org/wiki/Demilitarisierte\\_Zone\\_\(Informatik\)](https://de.wikipedia.org/wiki/Demilitarisierte_Zone_(Informatik))

[heise.de/ct/artikel/DMZ-selbst-gebaut-221656](https://heise.de/ct/artikel/DMZ-selbst-gebaut-221656)

### 10.6.1 Allgemein

- Dient dazu, ein sicheres "Zwischennetz" für Server, die sowohl von intern (GRÜN) als auch von extern (ROT) erreichbar sein sollen, zu schaffen
- Eine DMZ wird oft in oranger Farbe dargestellt.

### 10.6.2 Das 2-stufige Konzept

- wird in den IHK-Prüfungen bevorzugt
- ist aufwendiger:
  - 2 Firewalls mit je 2 Netzwerkkarten
  - 2 mal Regeln schreiben
  - Routing nicht vergessen!
- ist sicherer => doppelte "Verteidigungslinie"

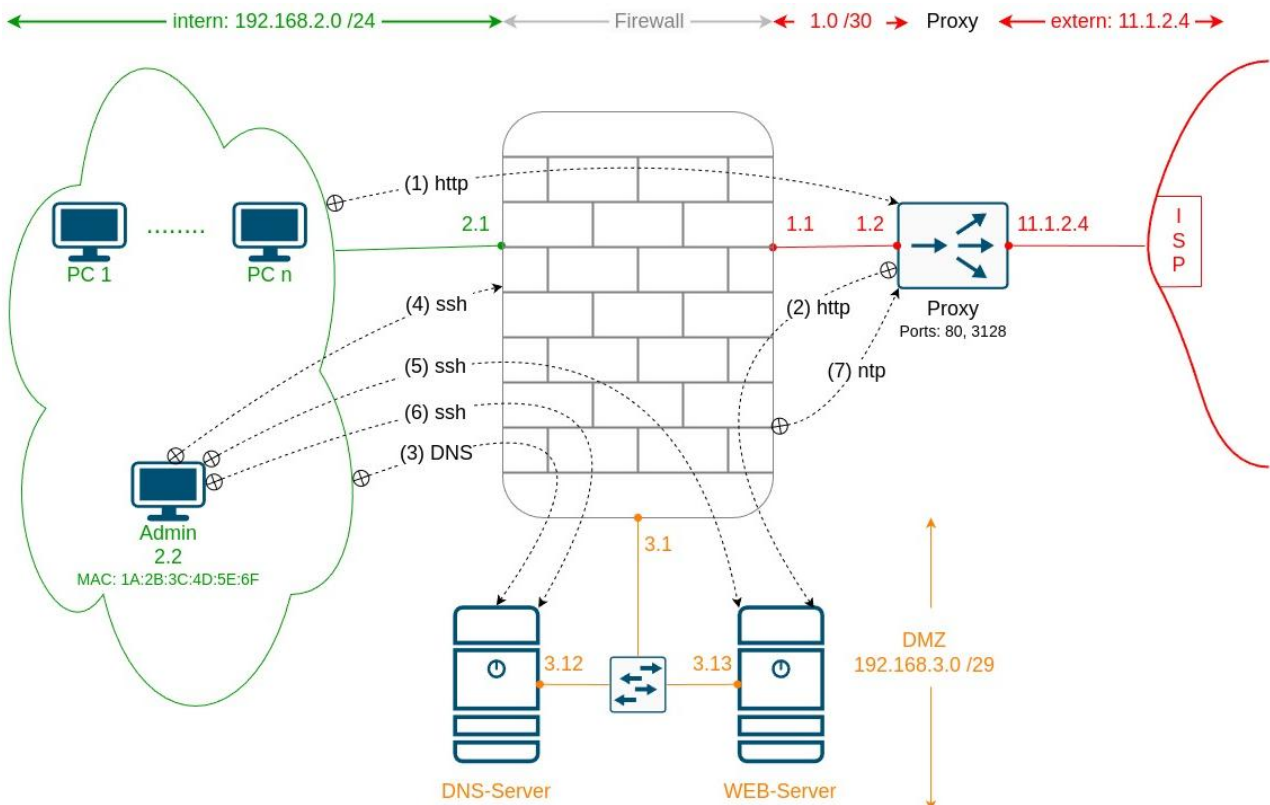
==>> Denken Sie an Burgen aus dem Mittelalter.

### 10.6.3 Das 1-stufige Konzept

- ist weniger aufwendig:
- nur eine Firewall mit 3 Netzwerkkarten
- nur einmalig Regeln schreiben
- ist unsicherer => einfache "Verteidigungslinie"

==>> Denken Sie an Stadttore aus dem Mittelalter.

## 10.7 Beispiel einer komplexeren Firewall mit einer DMZ



Füllt man die Tabelle entsprechend der Skizze aus, ergibt sich:

Regel	Input Output	TCP UDP ICMP	Quell-IP	Quell-MAC	Ziel-IP	Ziel-Port(s)	Kommentar
(1)	Forward	TCP	192.168.2.0/24		192.168.1.2	80,3128	LAN => Proxy
(2)	Forward	TCP	192.168.1.2		192.168.3.13	80	Proxy => WEB-Server
(3)	Forward	UDP	192.168.2.0/24		192.168.3.12	53	LAN => DNS-Server
(4)	Input	TCP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.2.1	22	Admin => Firewall
(5)	Forward	TCP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.3.13	22	Admin => WEB-Server
(6)	Forward	TCP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.3.12	22	Admin => DNS-Server
(7)	Output	UDP	192.168.1.1		192.168.1.2	123	Firewall => Proxy
(weitere)	Input	ICMP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.2.1	---	Ping
(last)			Alles andere wird verboten!				

Nutzt man die Datei /lptables\_Web/iptables.html und überträgt die Tabelle => Lösung:

	Direction	Protocol	Source IP	Source MAC	Target IP	Port(s)	Comment
<input checked="" type="checkbox"/>	FORWARD	TCP	192.168.2.0/24		192.168.1.2	80,3128	Regel (1)
<input checked="" type="checkbox"/>	FORWARD	TCP	192.168.1.2		192.168.3.13	80	Regel (2)
<input checked="" type="checkbox"/>	FORWARD	UDP	192.168.2.0/24		192.168.3.12	53	Regel (3)
<input checked="" type="checkbox"/>	INPUT	TCP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.2.1	22	Regel (4)
<input checked="" type="checkbox"/>	FORWARD	TCP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.3.13	22	Regel (5)
<input checked="" type="checkbox"/>	FORWARD	TCP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.3.12	22	Regel (6)
<input checked="" type="checkbox"/>	OUTPUT	UDP	192.168.1.1		192.168.1.2	123	Regel (7)
<input checked="" type="checkbox"/>	INPUT	ICMP	192.168.2.2	1A:2B:3C:4D:5E:6F	192.168.2.1		(weitere)

Unterhalb der Tabelle finden Sie auch wieder die Syntax für Iptables.

### 10.7.1 Auch hier wieder etwas „reverse engineering“

Das Resultat ist folgende Batch-Datei, die eine Linux-Maschine mit 3 Netzwerkkarten zu einer Firewall macht:

```
# start of user-defined rules
echo "Regel (1)"
$FT $MP -s 192.168.2.0/24 -d 192.168.1.2 --dports 80,3128 $R
echo "Regel (2)"
$FT -s 192.168.1.2 -d 192.168.3.13 --dport 80 $R
echo "Regel (3)"
$FU -s 192.168.2.0/24 -d 192.168.3.12 --dport 53 $R
echo "Regel (4)"
$IT $MAC 1A:2B:3C:4D:5E:6F -s 192.168.2.2 -d 192.168.2.1 --dport 22 $R
echo "Regel (5)"
$FT $MAC 1A:2B:3C:4D:5E:6F -s 192.168.2.2 -d 192.168.3.13 --dport 22 $R
echo "Regel (6)"
$FT $MAC 1A:2B:3C:4D:5E:6F -s 192.168.2.2 -d 192.168.3.12 --dport 22 $R
echo "Regel (7)"
$OU -s 192.168.1.1 -d 192.168.1.2 --dport 123 $R
echo "(weitere)"
$I $MAC 1A:2B:3C:4D:5E:6F -s 192.168.2.2 -d 192.168.2.1 $R
# end of user-defined rules
```

Das Skript kann wie oben beschrieben genutzt werden. Es gibt nur einen Unterschied:

Der PC braucht 3 Netzwerkkarten: 1. Netzwerkkarte => IP-Adresse: 92.168.1.1  
 2. Netzwerkkarte => IP-Adresse: 192.168.2.1  
 3. Netzwerkkarte => IP-Adresse: 192.168.3.1

# 11 Verschlüsselung

siehe Westermann Seite 527 ff

==>> Daten können heute teilweise verschlüsselt verarbeitet werden:

[de.wikipedia.org/wiki/Homomorphe\\_Verschl%C3%BCsslung](https://de.wikipedia.org/wiki/Homomorphe_Verschl%C3%BCsslung)

- Verschlüsselung für den Transport der Daten im Internet => ist anzuraten
- Verschlüsselung bei der Speicherung der Daten in der Cloud => auf jeden Fall!
- Verschlüsselung für die Verarbeitung der Daten in der Cloud => soll zukünftig möglich werden!
- für Wissenshungrige (Verschlüsselung lernen und selbst ausprobieren):

[de.wikipedia.org/wiki/CrypTool](https://de.wikipedia.org/wiki/CrypTool)

## 11.1 Unterscheidungsmerkmale

- symmetrisches Verfahren:  
[de.wikipedia.org/wiki/Symmetrisches\\_Kryptosystem](https://de.wikipedia.org/wiki/Symmetrisches_Kryptosystem)
- asymmetrisches Verfahren:  
[de.wikipedia.org/wiki/Asymmetrisches\\_Kryptosystem](https://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem)
- hybrides (kombinatorisches) Verfahren (asymmetrisches + symmetrisches Verfahren)
- Diffie-Hellman-Verfahren:  
[https://de.wikipedia.org/wiki/Elliptic\\_Curve\\_Cryptography](https://de.wikipedia.org/wiki/Elliptic_Curve_Cryptography)  
Elliptic Curve Cryptography  
  
siehe [de.wikipedia.org/wiki/Perfect\\_Forward\\_Secrecy](https://de.wikipedia.org/wiki/Perfect_Forward_Secrecy)
- Im Folgenden sind Alice und Bob Partner, Eve ist der/die/das „Bösewicht“, ersetzen Sie die 3 Buchstaben von „Eve“ mit Institutionen, die auch 3 Buchstaben im Namen tragen (NSA,...).
- Eve versucht, an die Daten zu kommen und sie zu manipulieren.

## 11.2 Symmetrisches Verfahren

eine Geldkassette als Beispiel



alternativ: ein Schlüsseltresor  
mit einem Zahlenschloss und zugehöriger PIN



Es werden 2 gleiche Schlüssel benötigt.

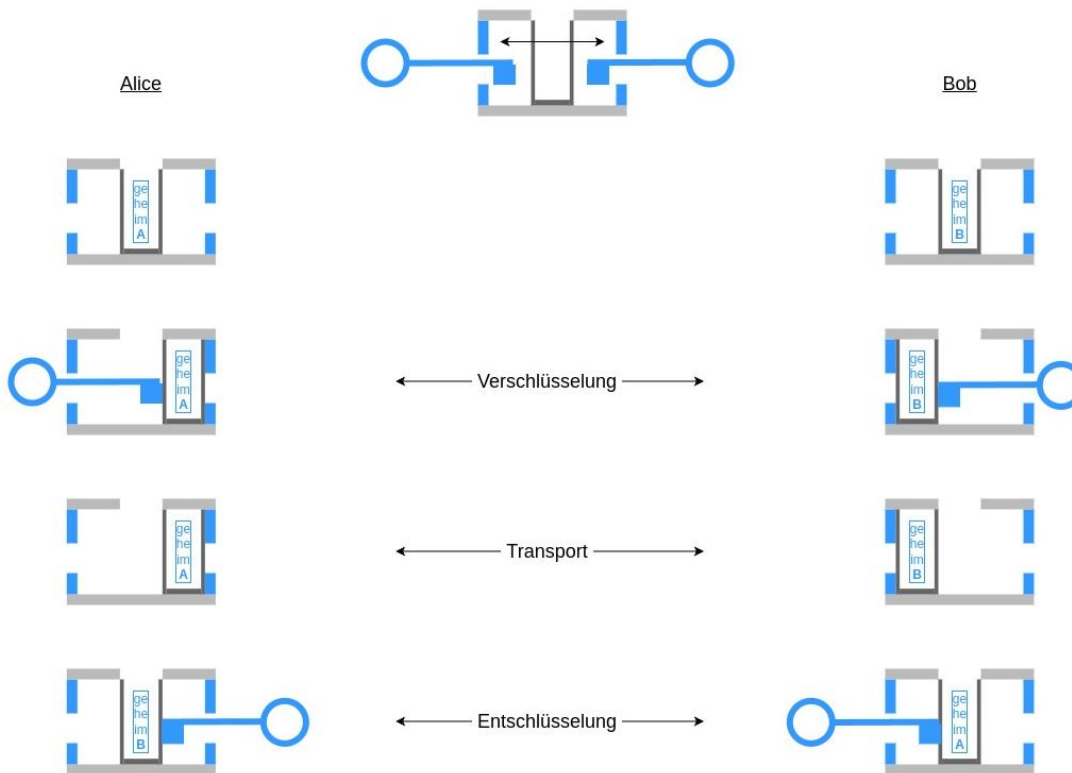
Beide Schlüssel sind gleichwertig => Ver- und Entschlüsseln ist mit beiden Schlüsseln möglich.

=> beide Partner benötigen einen identischen geheimen Schlüssel, in den folgenden Grafiken in blauer Farbe dargestellt

- sehr schnelles mathematisches Verfahren, 500- bis 1000-mal schneller, als asymmetrische Verschlüsselung.
- die Schlüssel sind relativ kurz ( $\geq 256$  bit)
- Ausnahme ist das „One-Time-Pad“ => einzige nicht zu knackende Verschlüsselung

[de.wikipedia.org/wiki/One-Time-Pad](https://de.wikipedia.org/wiki/One-Time-Pad)

### 11.2.1 Symmetrische Verschlüsselung in der Übersicht



- Wir haben ein bewegliches Element (das dunkelgraue „U“), das sich aus der Mitte heraus nach links und rechts verschieben lässt.
- Nur wenn sich das „U“ in der Mittelstellung befindet, können Geheimnisse hineingesteckt oder auch wieder entnommen werden.
- Auf beiden Seiten haben wir ein Schloss, in welches nur der geheime blaue Schlüssel passt.

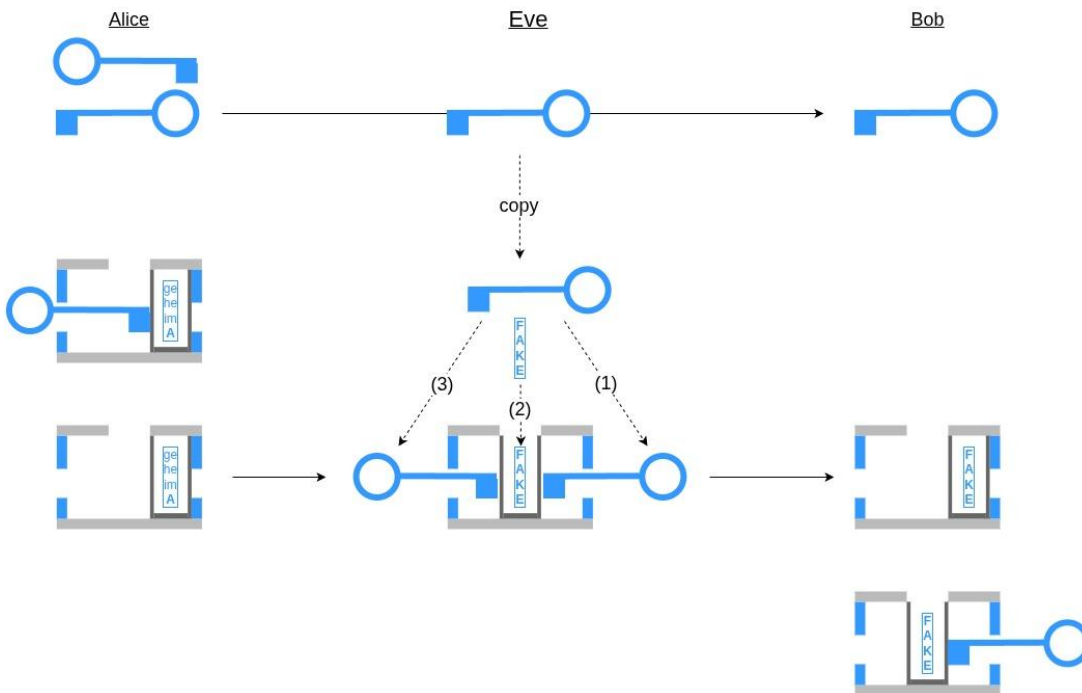
### 11.2.2 Verschlüsselung von Alice zu Bob

- Alice schiebt das „U“ mit dem blauen Schlüssel nach rechts  
=> sie verschlüsselt somit ihr Geheimnis „geheim A“
- Bob schiebt das „U“ mit dem blauen Schlüssel wieder in die Mitte  
=> er entschlüsselt dadurch das Geheimnis „geheim A“ von Alice

### 11.2.3 Verschlüsselung von Bob zu Alice

- Bob schiebt das „U“ mit dem blauen Schlüssel nach links  
=> er verschlüsselt somit sein Geheimnis „geheim B“
- Alice schiebt das „U“ mit dem blauen Schlüssel wieder in die Mitte  
=> sie entschlüsselt dadurch das Geheimnis „geheim B“ von Bob

## 11.2.4 Das Problem der Übergabe des symmetrischen Schlüssels



- Im Beispiel hat Alice einen blauen symmetrischen Schlüssel erzeugt.
- Eine 1:1-Kopie des Schlüssels muss an Bob übertragen werden.
- Wie bekommt Alice einen der beiden blauen Schlüssel zu Bob, ohne dass dieser von Eve kopiert werden kann?

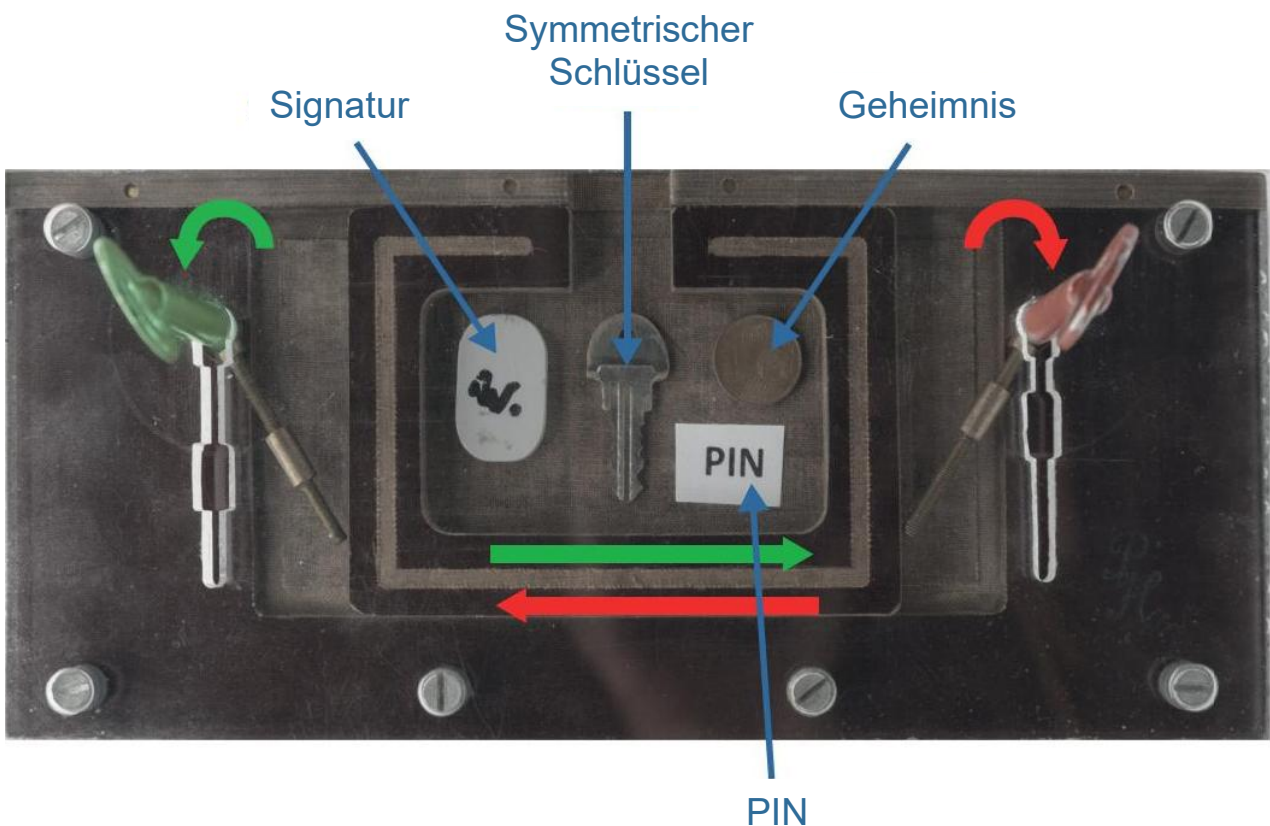
=> Bekommt Eve irgendwie den blauen Schlüssel in ihre Hände => Game Over! ;-)

## 11.3 Asymmetrisches Verfahren

==>> Der Empfänger des Geheimnisses muss das Schlüsselpaar erzeugen!

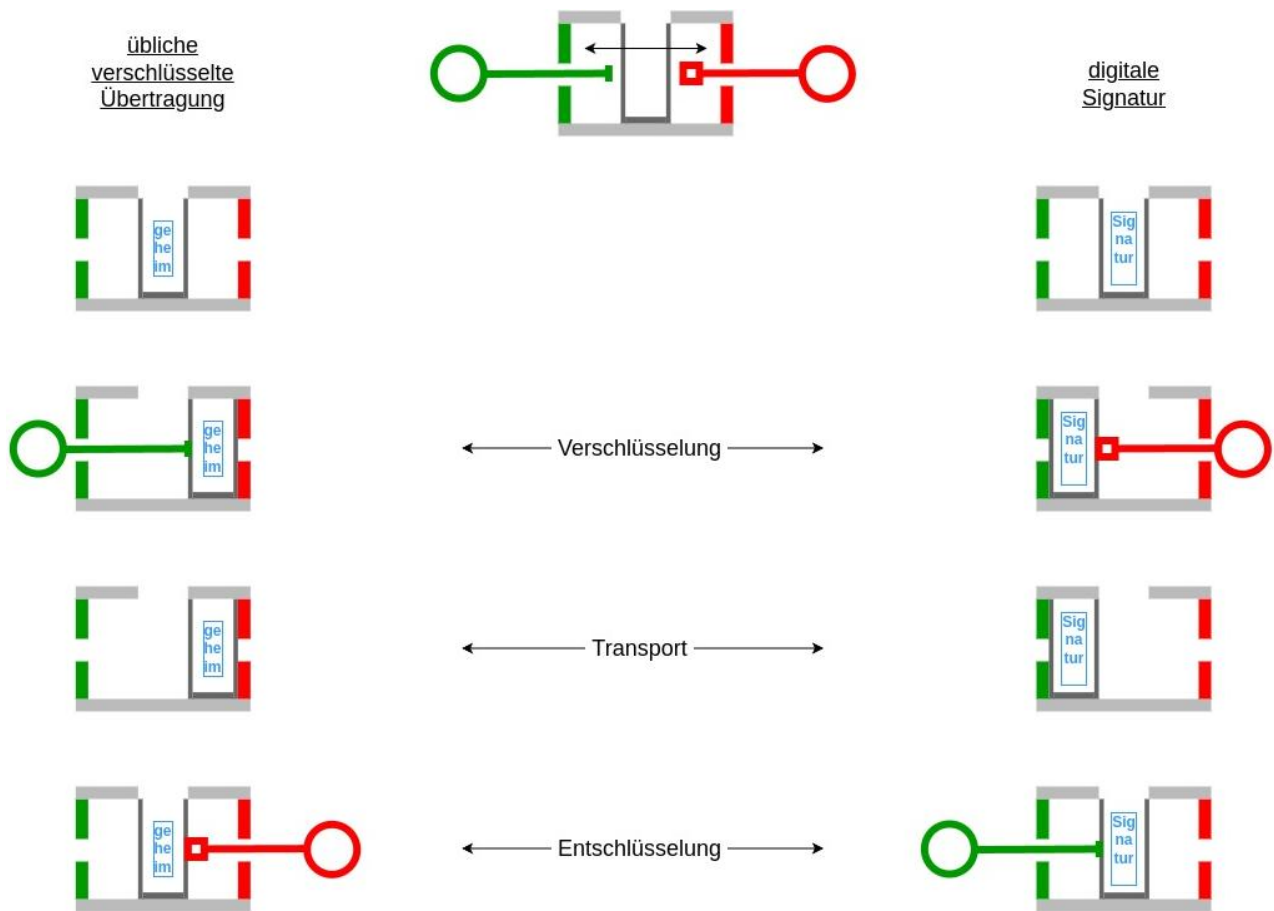
- Es werden 2 unterschiedliche Schlüssel benötigt.  
der rote Schlüssel => PRIVAT => wird niemals aus der Hand gegeben  
der grüne Schlüssel => ÖFFENTLICH => kann frei verteilt werden  
=> Was mit einem Schlüssel verschlüsselt wird, kann nur wieder mit dem anderen Schlüssel entschlüsselt werden!
- sehr langsames mathematisches Verfahren, 500- bis 1000-mal langsamer als die symmetrische Verschlüsselung
- Die Schlüssel sind sehr lang (> 3072 bit).

### 11.3.1 Beispiel einer selbst gebauten Mechanik



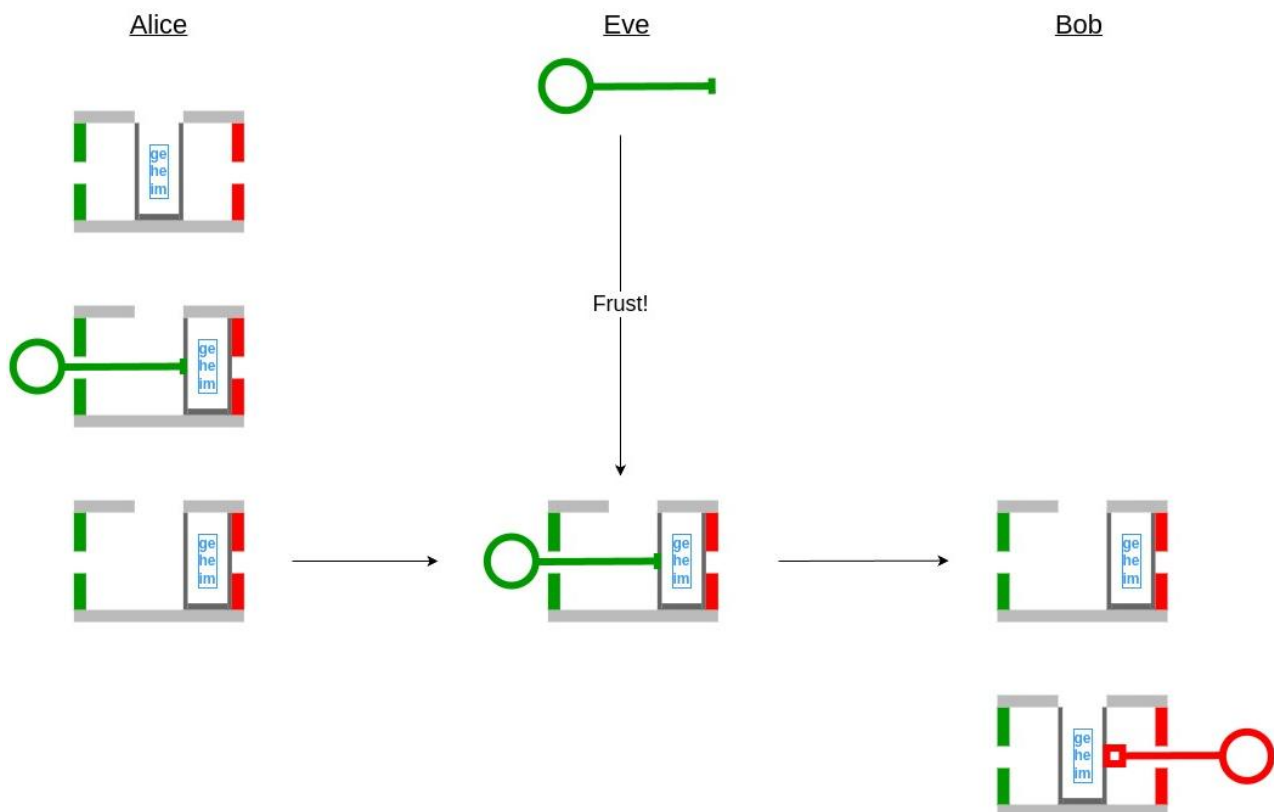
- wird der linke (grüne) Schlüssel gegen den Uhrzeigersinn gedreht => das Mittelteil bewegt sich nach rechts => die Öffnung verschließt sich
- wird der rechte (rote) Schlüssel im Uhrzeigersinn gedreht => das Mittelteil bewegt sich nach links => die Öffnung verschließt sich auch in diesem Fall
- nur in der Mittelstellung ist der Zugang möglich

### 11.3.2 Asymmetrisches Verfahren am Beispiel einer Grafik



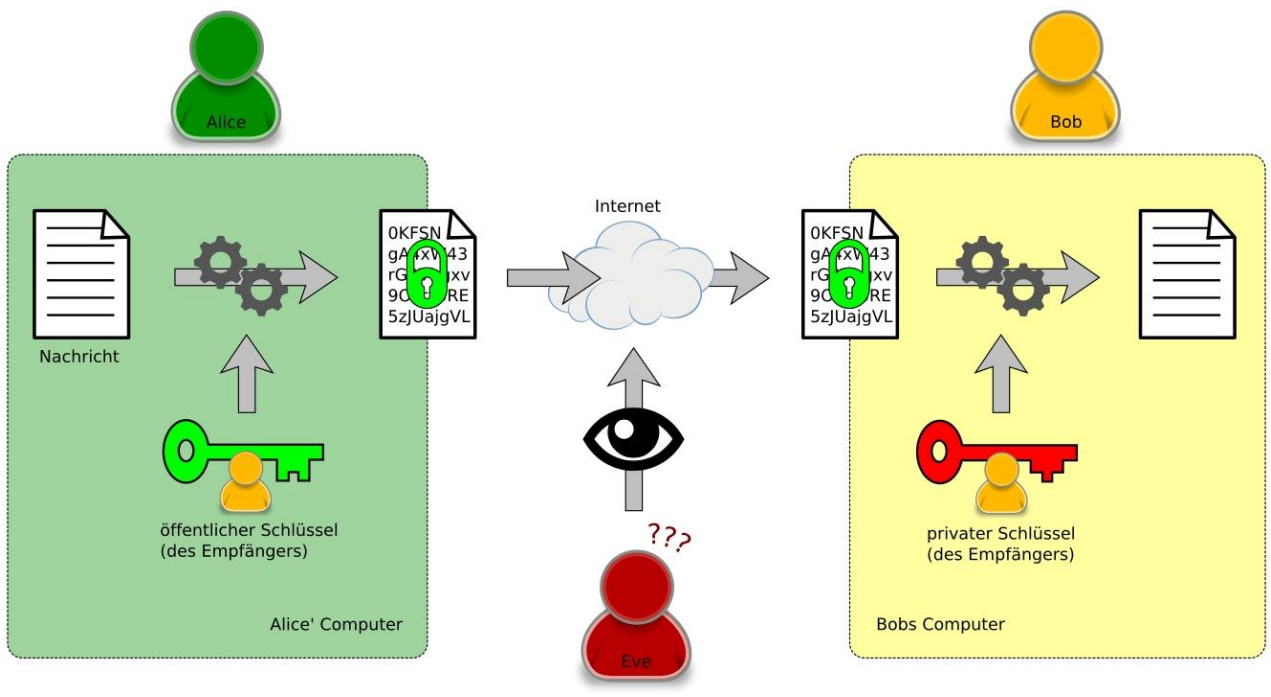
- Wir haben ein bewegliches Element (das dunkelgraue „U“), das sich aus der Mitte heraus nach links und rechts verschieben lässt.
- Nur wenn sich das „U“ in der Mittelstellung befindet, können Geheimnisse oder Signaturen hineingesteckt oder auch wieder entnommen werden.
- Auf beiden Seiten haben wir ein Schloss, in welches jeweils nur der grüne (öffentliche) Schlüssel bzw. der rote (private) Schlüssel passt.

### 11.3.3 Das Prinzip der Verschlüsselung

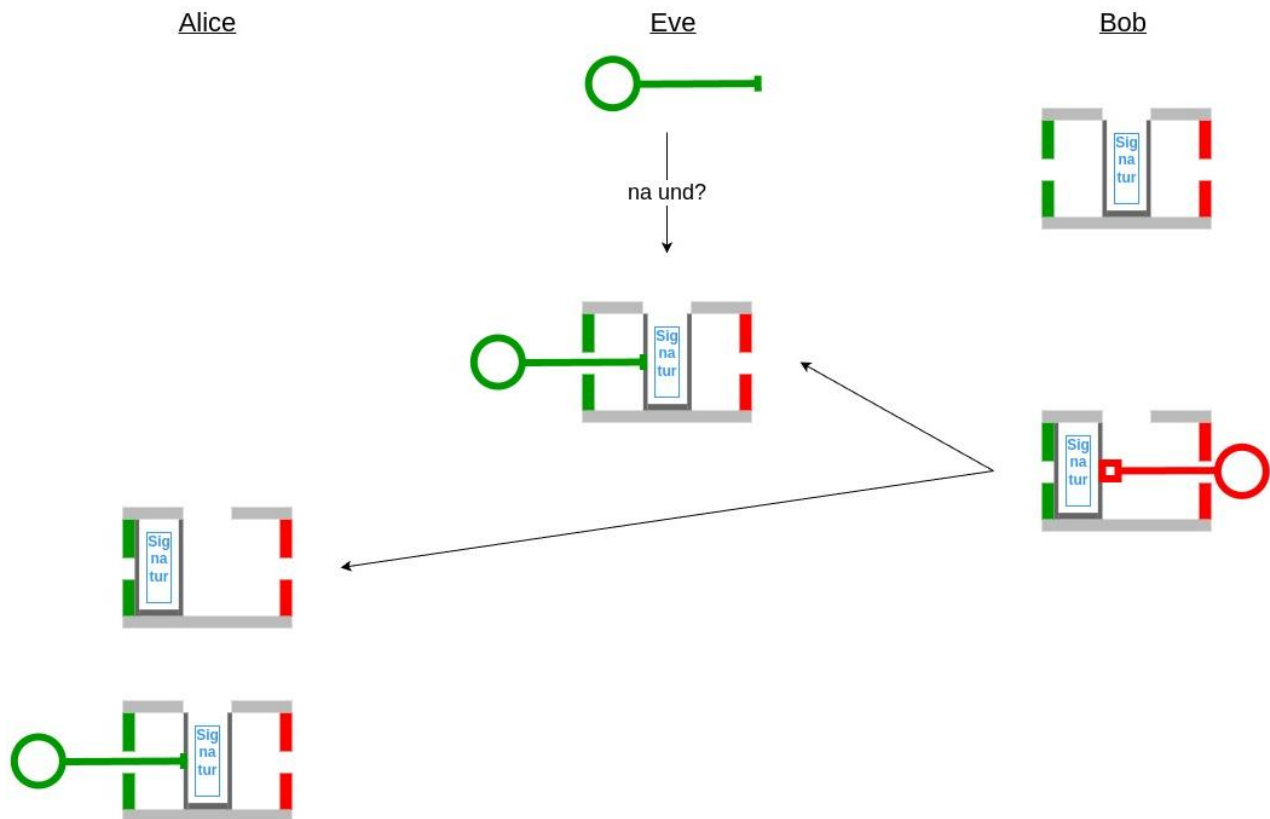


- Der Empfänger des Geheimnisses hat den roten und den grünen Schlüssel erzeugt, hier: Bob.
- Den roten Schlüssel gibt Bob nicht aus der Hand.
- Den grünen Schlüssel hat Bob veröffentlicht.  
=> Alice kann ihn benutzen.  
=> Eve hat sich auch eine Kopie des grünen Schlüssels besorgt.
- Das „U“ ist in der Mitte => ein Geheimnis kann hineingesteckt werden („geheim“ von Alice).
- Alice verschiebt mit dem grünen Schlüssel das „U“ nach rechts => die Öffnung verschließt sich.
- Eve hat keine Chance, mit dem grünen Schlüssel das „U“ in die Mitte zu bekommen.
- Bob verschiebt mit dem roten Schlüssel das „U“ nach links bis zur Mitte => die Öffnung ist wieder zugänglich => Bob kann das „geheim“ von Alice entnehmen.

Eine alternative Grafik eines ehemaligen Teilnehmers:



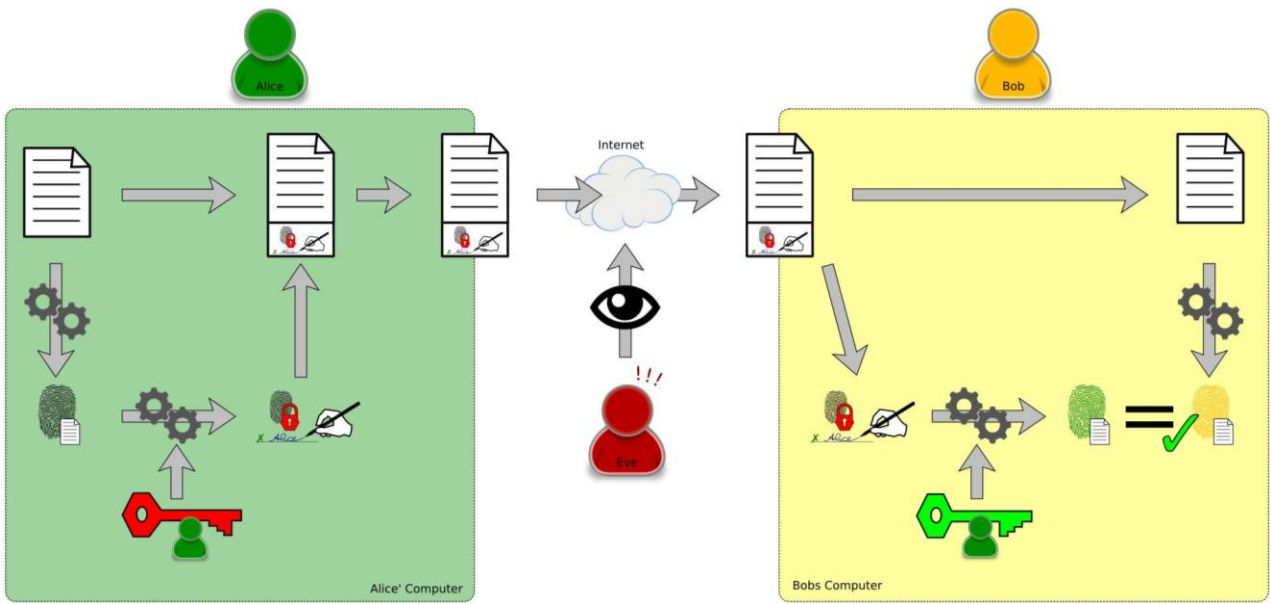
### 11.3.4 Das Prinzip der Signatur



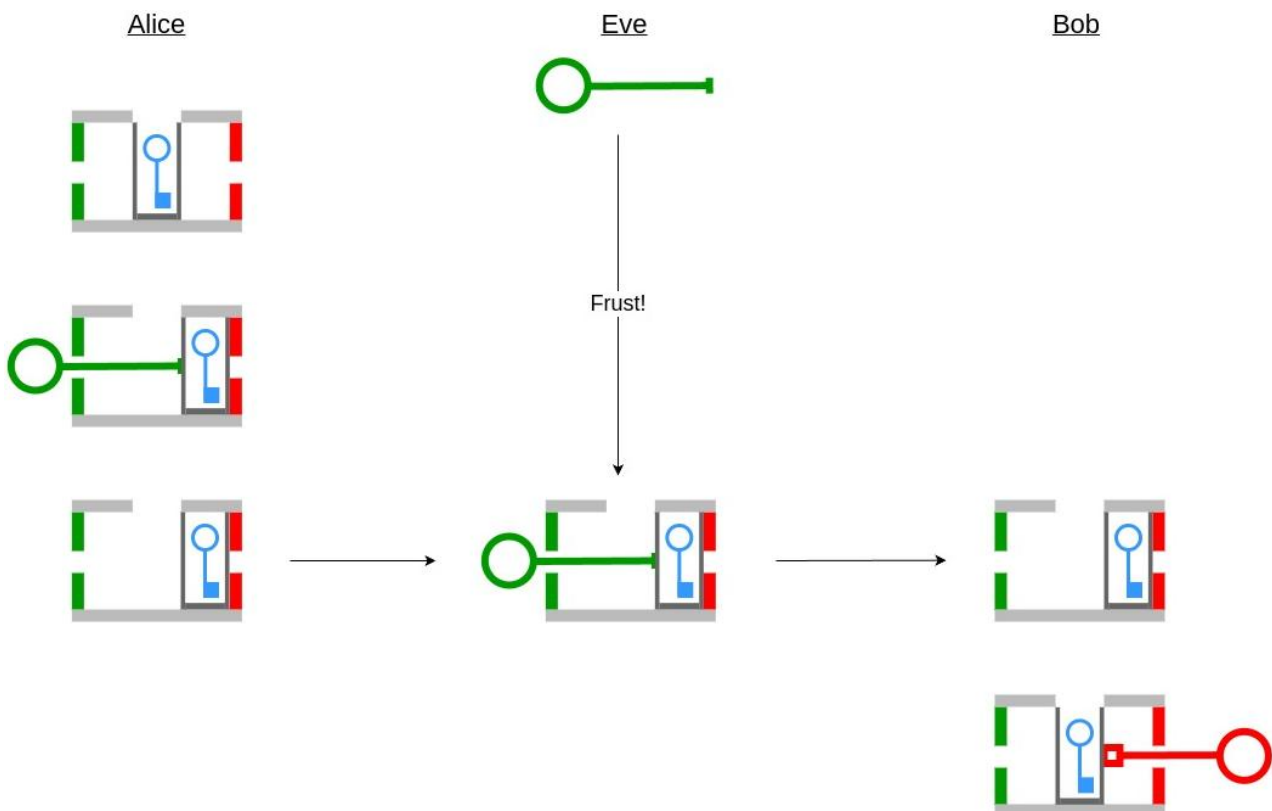
=> Die asymmetrische Verschlüsselung funktioniert auch andersherum.

- Der Sender der Signatur hat den roten und den grünen Schlüssel erzeugt, hier: Bob.
- Den roten Schlüssel gibt Bob nicht aus der Hand.
- Den grünen Schlüssel hat Bob veröffentlicht.
  - => Alice kann ihn benutzen.
  - => Eve hat sich auch eine Kopie des grünen Schlüssels besorgt.
  - => Was mit einem Schlüssel verschlüsselt wird, kann nur wieder mit dem anderen Schlüssel entschlüsselt werden! => erinnern Sie sich?
- Das „U“ ist in der Mitte => Eine Signatur kann hineingesteckt werden („Signatur“ von Bob).
- Bob verschiebt mit dem roten Schlüssel das „U“ nach links => die Öffnung verschließt sich.
- Eve und Alice können mit dem grünen Schlüssel das „U“ nach rechts, bis zur Mitte verschieben => die Öffnung ist wieder zugänglich => Eve und Alice können sich die Signatur von Bob ansehen.

Auch dazu eine alternative Grafik eines ehemaligen Teilnehmers: ( Siehe nächste Seite )

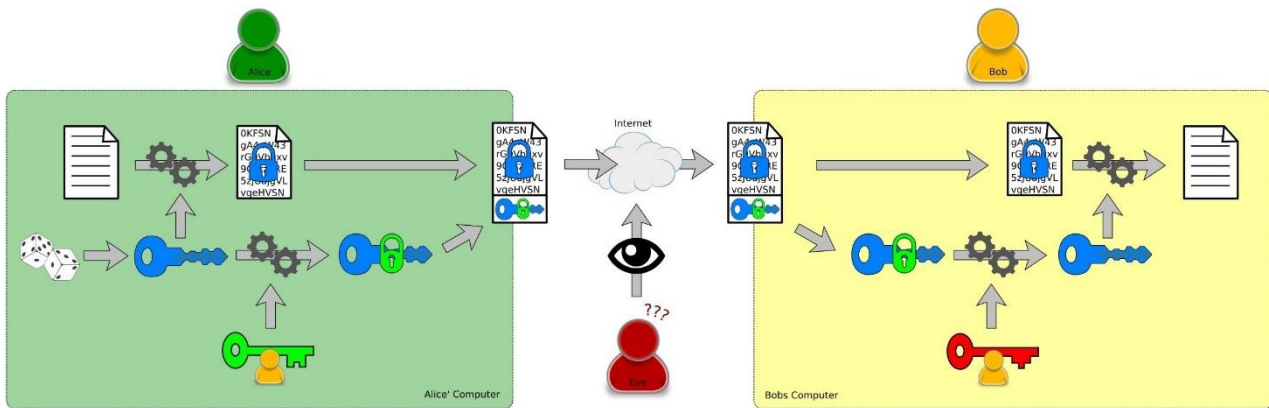


### 11.3.5 Lösung: Übergabe des symmetrischen Schlüssels



- Nutzen Sie die asymmetrische Verschlüsselung nicht zur Übergabe eines Geheimnisses, sondern zur Übergabe des symmetrischen Schlüssels.  
=> Der symmetrische Schlüssel kann sicher übertragen werden.
- Nutzen Sie anschließend die viel schnellere symmetrische Verschlüsselung.

Auch dazu eine alternative Grafik eines ehemaligen Teilnehmers: ( Siehe nächste Seite )



- In diesem Beispiel werden die Anmeldedaten für ein WLAN mit einem zufälligen symmetrischen Schlüssel verschlüsselt.
- Der symmetrische Schlüssel wird anschließend asymmetrisch verschlüsselt.
- Bei Bob wird der symmetrische Schlüssel entschlüsselt.
- Bob hat den symmetrischen Schlüssel sicher erhalten.
- Bob kann die Anmeldedaten mit dem symmetrischen Schlüssel entschlüsseln.

### 11.3.6 Erklärung der asymmetrischen Verschlüsselung mit trivialer Mathematik

- Alice möchte Bob eine verschlüsselte Nachricht senden.
- Als Empfänger der Nachricht muss Bob das Schlüsselpaar erstellen.
- Bob rechnet:
  - $4 * 7 \text{ modulo } 27 = 1$
  - => das Zahlenpaar 4 und 27 sind der grüne, öffentliche Schlüssel
  - => werden an Alice zum Verschlüsseln gegeben
  - => das Zahlenpaar 7 und 27 sind der rote private Schlüssel
  - => bleiben bei Bob zum Entschlüsseln
- Verschlüsselung: Alice möchte „o“ an Bob senden und rechnet:
  - Buchstabe „o“ => 15. Buchstabe im Alphabet
  - $15 * 4 \text{ modulo } 27 = 6$
- Die 6 (das Chifftrat) wird an Bob gesendet.
  - Bob empfängt die 6 (das Chifftrat) und rechnet:
    - $6 * 7 \text{ modulo } 27 = 15$
    - der 15. Buchstabe im Alphabet = „o“
- Für das „k“ probieren Sie doch bitte selbst einmal beide Richtungen aus.

## **11.4 Besonders drei Begriffe spielen eine außergewöhnliche Rolle**

siehe Westermann Seite 532

### **11.4.1 Authentizität**

Frage: Bist Du wirklich derjenige, für den Du Dich ausgibst?

[de.wikipedia.org/wiki/Shared Secret](https://de.wikipedia.org/wiki/Shared_Secret)

Zertifikat: siehe Westermann Seite 294

[de.wikipedia.org/wiki/Digitales Zertifikat](https://de.wikipedia.org/wiki/Digitales_Zertifikat)

[de.wikipedia.org/wiki/Public-Key-Zertifikat](https://de.wikipedia.org/wiki/Public-Key-Zertifikat)

### **11.4.2 Integrität**

Frage: Wurden die Daten auf dem Transportweg verändert?

Hashfunktion:

siehe Westermann Seite 293

[de.wikipedia.org/wiki/Hashfunktion](https://de.wikipedia.org/wiki/Hashfunktion)

Digitale Signatur:

siehe Westermann Seite 294

[de.wikipedia.org/wiki/Digitale Signatur](https://de.wikipedia.org/wiki/Digitale_Signatur)

### **11.4.3 Vertraulichkeit**

Frage: Sind die Daten sicher, können Dritte die Daten lesen?

Verschlüsselungsalgorithmen: siehe Westermann Seite 528

[de.wikipedia.org/wiki/Verschlüsselung](https://de.wikipedia.org/wiki/Verschlüsselung)

## 11.5 Zusammenfassung Verschlüsselung

Ist die Verschlüsselung nicht zu knacken?

Antwort:

- Haben die Verschlüsselungsalgorithmen eine (gewollte) Hintertür?
- Wurden die Verschlüsselungsalgorithmen fehlerfrei in die jeweilige Software implementiert?

Echte Zufallszahlen:

[de.wikipedia.org/wiki/Zufallszahlengenerator](https://de.wikipedia.org/wiki/Zufallszahlengenerator)

[random.org/analysis/](https://random.org/analysis/)

Brute-Force:

[de.wikipedia.org/wiki/Brute-Force-Methode](https://de.wikipedia.org/wiki/Brute-Force-Methode)

- zur Erinnerung: Bitcoin-Mining:
  - mit normalem PC => sehr langsam
  - mit der Grafikkarte => schneller
  - mit ASICs => sehr viel schneller

[de.wikipedia.org/wiki/Anwendungsspezifische integrierte Schaltung](https://de.wikipedia.org/wiki/Anwendungsspezifische_integrierte_Schaltung)

- einzige "unknackbare" Verschlüsselung: das One-Time-Pad:

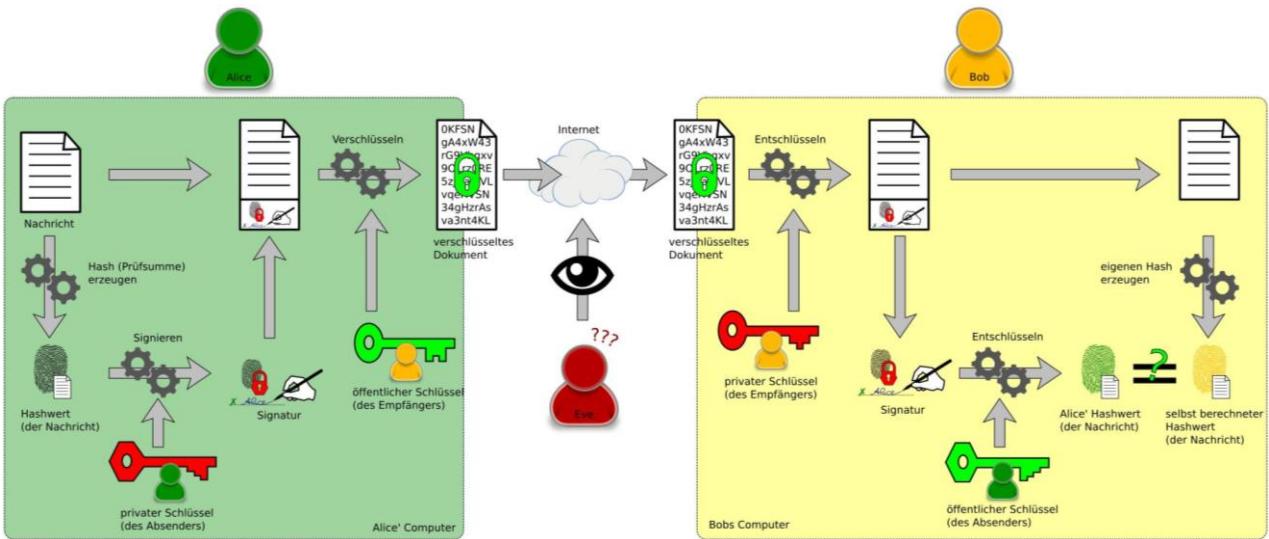
[de.wikipedia.org/wiki/One-Time-Pad](https://de.wikipedia.org/wiki/One-Time-Pad)

[de.wikipedia.org/wiki/Exklusiv-Oder-Gatter](https://de.wikipedia.org/wiki/Exklusiv-Oder-Gatter)

### 11.5.1 Erste Herausforderung

Wer gern noch mehr tüfteln möchte und die Komplexität der Verschlüsselung nicht scheut:

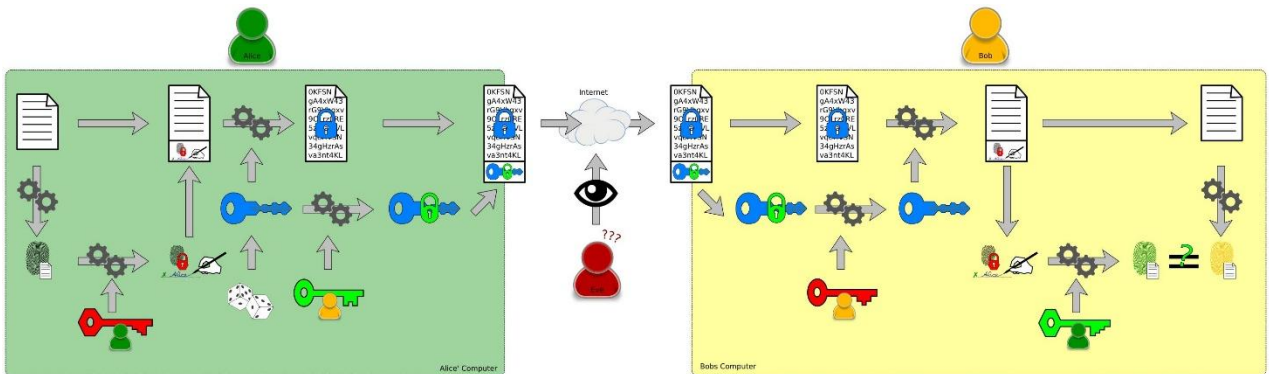
- Nachricht mit Signatur und asymmetrischer Verschlüsselung



### 11.5.2 Zweite Herausforderung

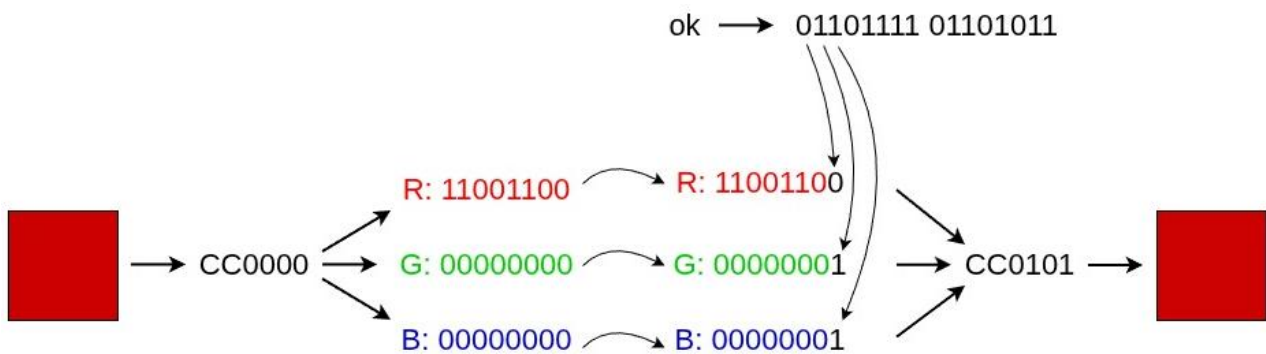
Und jetzt raucht der Kopf endgültig. ;-)

- Nachricht mit Signatur und hybride Verschlüsselung mit erster Nachricht



## 11.6 Steganographie

versteckte Informationen in alltäglichen unauffälligen Daten



- Ein originaler Bildpunkt besteht aus  $3 * 8$  bit (Rot, Grün, Blau).
- Das unbedeutendste Bit (least significant bit) wird durch die Information verändert.
- Das Bild ist hinterher verändert, der Unterschied aber kaum zu sehen.
- Ein Bildpunkt kann 3 bit „verstecken“.
- ideal: die Kombination aus Verschlüsselung und Steganographie

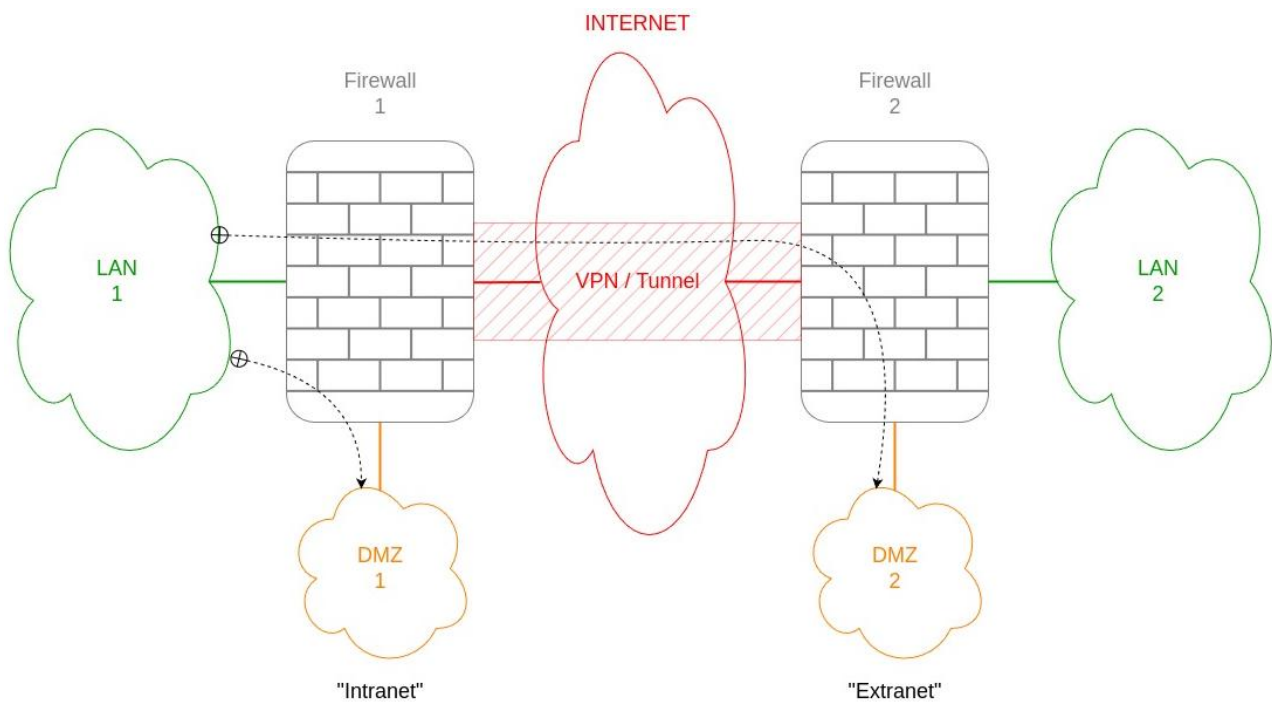
## 12 Intranet und Extranet im VPN (Virtuelles Privates Netzwerk)

siehe Westermann Seite 598

[wiki.securepoint.de/UTM/VPN/%C3%9Cbersicht](http://wiki.securepoint.de/UTM/VPN/%C3%9Cbersicht)

[de.wikipedia.org/wiki/Virtual Private Network](http://de.wikipedia.org/wiki/Virtual_Private_Network)

[de.wikipedia.org/wiki/IPsec](http://de.wikipedia.org/wiki/IPsec)



Eine sehr moderne und praktische Lösung: [de.wikipedia.org/wiki/WireGuard](http://de.wikipedia.org/wiki/WireGuard)

### 12.1 Die Begriffe Intranet und Extranet aus dem Blickwinkel von LAN 1

Besteht zwischen beiden Firewalls eine VPN-Verbindung, dann gilt:

- Die DMZ 1 (die eigene DMZ) wird als Intranet bezeichnet (schwarz gestrichelte Linie).
- Die DMZ 2 (die fremde DMZ, die DMZ von LAN 2) wird als Extranet bezeichnet (schwarz gestrichelte Linie).
- Für LAN 2 sind Intranet und Extranet nicht eingezeichnet, die Übersichtlichkeit ginge sonst verloren.

## 12.2 Transportmodus vs. Tunnelmodus

[heise.de/security/artikel/VPN-Knigge-270796.html?seite=all](https://heise.de/security/artikel/VPN-Knigge-270796.html?seite=all)

- folgende Ausgangssituation: Alice aus Mühlenbeck möchte geheime Daten an Bob in Berlin senden

### 12.2.1 Transportmodus (als Metapher)

- Alice packt geheime Daten in einen Briefumschlag.
- Der Briefumschlag wird adressiert (Absender: Alice aus Mühlenbeck, Empfänger: Bob in Berlin).
- Anschließend wird der Briefumschlag mit der Post verschickt.
- Was sieht Eve?  
Alice aus Mühlenbeck schickt Bob in Berlin geheime Daten.
- Was sieht Eve nicht?  
den Inhalt des Briefumschlags (die geheimen Daten)

### 12.2.2 Tunnelmodus (als Metapher)

- Alice packt geheime Daten in einen Briefumschlag.
- Der Briefumschlag wird adressiert (Absender: Alice aus Mühlenbeck, Empfänger: Bob in Berlin).
- Anschließend wird der Briefumschlag mit einem Kurier verschickt, der von Mühlenbeck nach Berlin fährt.
- Was sieht Eve?  
einen Kurier aus Mühlenbeck, der nach Berlin fährt
- Was sieht Eve nicht?  
den Absender (Alice aus Mühlenbeck)  
den Empfänger (Bob in Berlin)  
den Inhalt des Briefumschlags (die geheimen Daten)

# 13 Cloud

siehe Westermann Seite 624

[de.wikipedia.org/wiki/Cloud Computing](http://de.wikipedia.org/wiki/Cloud_Computing)

[de.wikipedia.org/wiki/Everything as a Service#Infrastructure as a Service \(IaaS\)](http://de.wikipedia.org/wiki/Everything_as_a_Service#Infrastructure_as_a_Service_(IaaS))

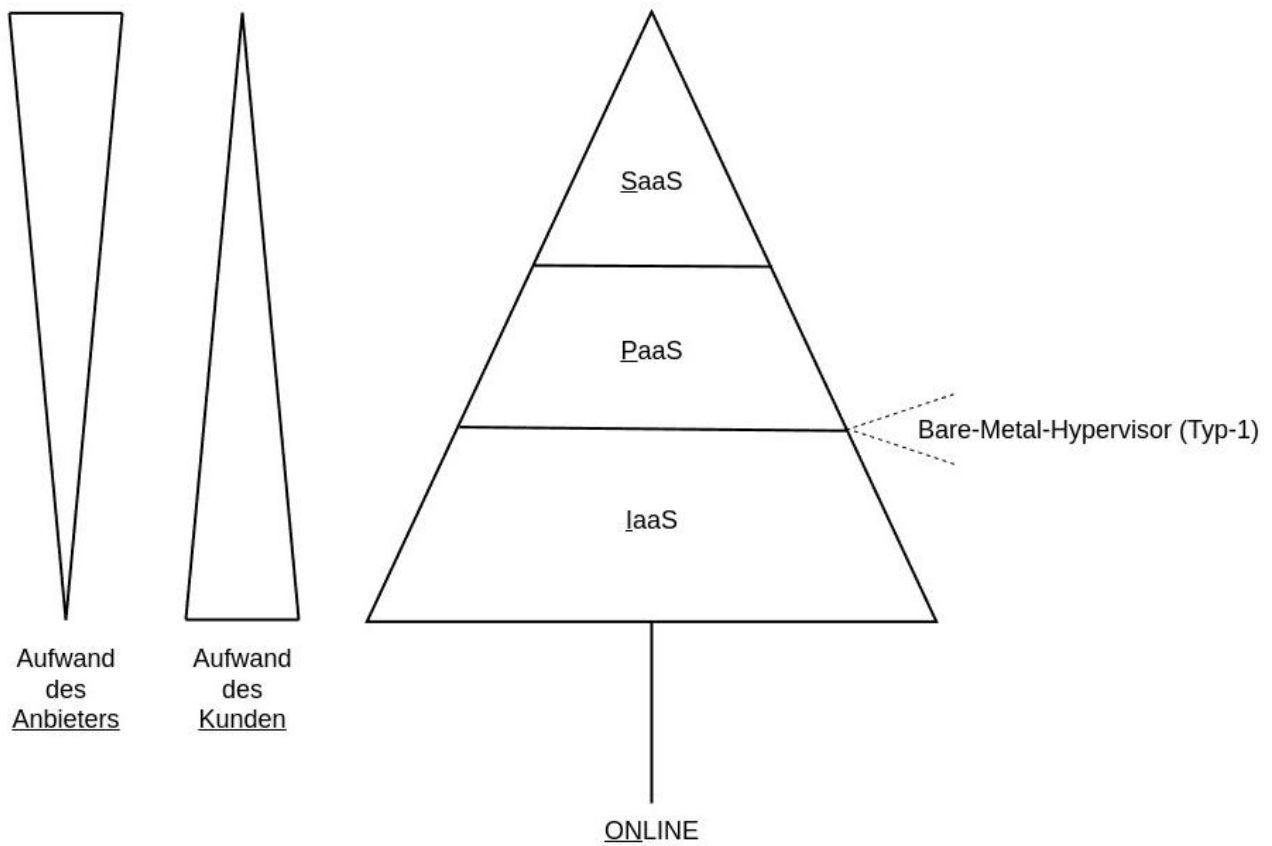
[de.wikipedia.org/wiki/Platform as a Service](http://de.wikipedia.org/wiki/Platform_as_a_Service)

[de.wikipedia.org/wiki/Software as a Service](http://de.wikipedia.org/wiki/Software_as_a_Service)

[ionos.de/digitalguide/server/knowhow/iaas-infrastructure-as-a-service](http://ionos.de/digitalguide/server/knowhow/iaas-infrastructure-as-a-service)

[ionos.de/digitalguide/server/knowhow/paas-platform-as-a-service](http://ionos.de/digitalguide/server/knowhow/paas-platform-as-a-service)

[ionos.de/digitalguide/server/knowhow/saas-software-as-a-service-im-ueberblick-vor-und-nachteile](http://ionos.de/digitalguide/server/knowhow/saas-software-as-a-service-im-ueberblick-vor-und-nachteile)



## 13.1 Sicherheit

Ohne einen Alu-Hut zu tragen, heißt (Public-) Cloud:

- Meine Daten befinden sich auf unbekanntem Servern irgendwo auf der Welt.
- Die Anbieter versichern natürlich, dass die Daten in Deutschland / EU verbleiben. ;-)

Es gibt heute erste Ansätze, verschlüsselte Daten zu verarbeiten, bis dahin gilt:

- Daten müssen vor der Verarbeitung in der CPU entschlüsselt werden.
- Die Daten liegen somit in Klarschrift vor.
- Daten können in der CPU leicht kopiert werden.
- Die unterstrichenen Buchstaben von oben nach unten gelesen ergeben einen flachen Witz.

## 13.2 Aufwand für die Nutzung einer Cloud

### 13.2.1 Linker Keil ("Anbieter")

- Kunde mietet IaaS => geringer Aufwand => geringer Gewinn
- Kunde mietet PaaS => mäßiger Aufwand => mäßiger Gewinn
- Kunde mietet SaaS => hoher Aufwand => höchster Gewinn

### 13.2.2 Rechter Keil ("Kunde")

- Kunde mietet IaaS => hoher Aufwand, viele Freiheiten => preiswert
- Kunde mietet PaaS => mäßiger Aufwand, eingeschränkte Freiheiten => teurer
- Kunde mietet SaaS => geringer Aufwand, wenige Freiheiten => noch teurer

## **13.3 Beispiele**

### **13.3.1 SaaS**

- Web-Mailing
- Office 365
- MS-Teams

### **13.3.2 PaaS**

- virtualisierte Betriebssysteme
- XAMPP:
  - Apache => Webserver
  - MariaDB => Datenbank
  - Perl => Skriptsprache
  - PHP => Skriptsprache

### **13.3.3 IaaS**

- Serverraum
- Spannungsversorgung (Notstrom, USV)
- Klimatechnik
- Server-Racks
- Server-Hardware (Switches, Backup-Systeme)

==>> Trennschicht zwischen PaaS und IaaS: Bare-Metal-Hypervisor (Typ 1)

[de.wikipedia.org/wiki/Hypervisor](https://de.wikipedia.org/wiki/Hypervisor)

## 13.4 Auflistung der Cloud-Typen

(Als Metapher kann man auch an die eigene „Stromversorgung“ zu Hause denken.)

### 13.4.1 Public Cloud

[de.wikipedia.org/wiki/Cloud\\_Computing#Public\\_Cloud\\_-\\_die\\_oeffentliche\\_Rechnerwolke](https://de.wikipedia.org/wiki/Cloud_Computing#Public_Cloud_-_die_oeffentliche_Rechnerwolke)

- Der „Strom“ stammt von irgendwo her (Kohle, Kernkraft, Photovoltaik, Wind, Wasser), egal, was der Anbieter auch immer verspricht.

### 13.4.2 Hybride Cloud

[de.wikipedia.org/wiki/Cloud\\_Computing#Hybrid\\_Cloud\\_%E2%80%93\\_die\\_hybride\\_Rechnerwolke](https://de.wikipedia.org/wiki/Cloud_Computing#Hybrid_Cloud_%E2%80%93_die_hybride_Rechnerwolke)

- Der „Strom“ stammt von der eigenen Photovoltaik vom Dach und zusätzliche Leistung wird vom „Stromanbieter“ genutzt (siehe oben).

### 13.4.3 Community Cloud

[de.wikipedia.org/wiki/Cloud\\_Computing#Community\\_Cloud\\_%E2%80%93\\_die\\_gemeinschaftliche\\_Rechnerwolke](https://de.wikipedia.org/wiki/Cloud_Computing#Community_Cloud_%E2%80%93_die_gemeinschaftliche_Rechnerwolke)

- Eine Gemeinschaft baut eine große Photovoltaikanlage und / oder nutzt einen Windpark für die eigene Energieversorgung.

### 13.4.4 Private Cloud

[de.wikipedia.org/wiki/Cloud\\_Computing#Private\\_Cloud\\_%E2%80%93\\_die\\_private\\_Rechnerwolke](https://de.wikipedia.org/wiki/Cloud_Computing#Private_Cloud_%E2%80%93_die_private_Rechnerwolke)

- Der „Strom“ stammt ausschließlich von der eigenen Photovoltaikanlage vom Dach, man lebt sozusagen autark.

## 14 Phishing

[de.wikipedia.org/wiki/Phishing](https://de.wikipedia.org/wiki/Phishing)

siehe /Filius\_Szenen/14\_Phishing\_DHCP\_Server\_Teams.flv

### 14.1 Erklärung zu 14\_Phishing\_DHCP\_Server\_Teams.flv

- Der Server ATTACKE beinhaltet einen DHCP-, einen DNS- und einen Webserver.
- Die Switches S\_A\_1 und S\_A\_2 erzeugen eine gewollte Latenz.
- Das Laptop daneben soll wie alle Laptops im LAN des Unternehmens per DHCP konfiguriert werden.
- Beim Start wird durch S\_A\_1 und S\_A\_2 die Anfrage an den DHCP-Server des Unternehmens absichtlich verzögert.
- Das DHCP-Offer des DHCP-Servers auf ATTACK ist schneller und wird ungeprüft vom Client angenommen.
- Inhalt des Offers:  
Eine „Fake-DNS-IP-Adresse“ des DNS-Servers ATTACKE.
- Bei einer Anfrage des Clients (Laptop) nach teams.com gibt der DNS-Server von ATTACKE eine Antwort auf sich selbst zurück.
- Der Client baut eine Verbindung zum vermeintlichen Webserver von teams.com auf, dieser läuft aber auch auf ATTACKE, in Form eines „Fake-teams-Webservers“.
- Der Anwender am Laptop merkt nicht, dass er auf einer falschen Webseite „gelandet“ ist.

## 15 ARP-Spoofing

[de.wikipedia.org/wiki/ARP-Spoofing](https://de.wikipedia.org/wiki/ARP-Spoofing)

siehe /Filius\_Szenen/15\_ARP-Spoofing.flv

### 15.1 Erklärung zu 15\_ARP-Spoofing.flv

- Am PC 192.168.1.3 befindet sich „the-man-in-the-middle“ und hat dort „forwarding“ aktiviert (nimmt jedes Datenpaket auf, analysiert es und schickt es erst dann an sein eigentliches Ziel).
- Verschiedet wird ein „arp-req“ mit der IP-Adresse 192.168.1.1 an den Server mit der IP-Adresse 192.168.1.2, jedoch mit der eigenen MAC-Adresse.
- Verschiedet wird ein „arp-req“ mit der IP-Adresse 192.168.1.2 an den Server mit der IP-Adresse 192.168.1.1, jedoch auch wieder mit der eigenen MAC-Adresse.
- Resultat:  
Alle Datenpakete zwischen 192.168.1.1 und 192.168.1.2 machen einen Umweg über 192.168.1.3.

### 15.2 Schutz gegen ARP-Spoofing

- Die SPI-Personal-Firewall auf allen Maschinen im LAN muss immer eingeschaltet bleiben!
- Sie verhindert, dass ein ARP-REPLY angenommen wird, obwohl kein ARP-REQUEST gestellt wurde.

## 16 E-Mailing

[de.wikipedia.org/wiki/E-Mail](https://de.wikipedia.org/wiki/E-Mail)

siehe /Filius\_Szenen/16\_Alice\_Mail\_Bob\_intern.fls

### 16.1 Erklärung zu 16\_Alice\_Mail\_Bob\_intern.fls

- SMTP wird zum Versenden der E-Mail verwendet.
- POP3 wird zur Abholung der E-Mail verwendet.
- Die E-Mail wird nach Abholung auf dem E-Mail-Server gelöscht => es erfolgt ein „Verschieben“ der E-Mail.

### 16.2 Alternative zu POP3

- IMAP, IMAPS neuerdings JMAP
- Die E-Mail kann als Kopie vom Server geholt werden und dort im Original verbleiben oder wie bei POP3 „verschoben“ werden.

**Geschafft!**

## 17 Anhang

### Ein Blick über den Tellerrand

Wenn einem Mathematik nicht fremd, man aber nicht tief genug in diese Naturwissenschaft eingetaucht ist, kann man Probleme mit den vorhandenen Mitteln versuchen zu lösen. Eine Idee wäre hierbei, die „sukzessive Approximation“ (Iteration oder auch schrittweise Annäherung genannt) anzuwenden.

Man denke dabei an eine Balkenwaage:

[de.wikipedia.org/wiki/Balkenwaage](http://de.wikipedia.org/wiki/Balkenwaage)

In der einen Waagschale liegt ein Objekt unbekannter Masse. In die andere Waagschale legt man „Gewichte“ mit 1 kg, 0,1 kg, 0,01 kg, 0,001 kg ....., bis sich beide Waagschalen nahezu im Gleichgewicht befinden.

Auf diesem Wege kann man sich auch dem „Goldenen Schnitt“ nähern:

[de.wikipedia.org/wiki/Goldener\\_Schnitt](http://de.wikipedia.org/wiki/Goldener_Schnitt)

$a / b = b / c$  |  $c$  durch  $a + b$  ersetzen

$a / b = b / (a + b)$  | über Kreuz multiplizieren

$b^2 = a * (a + b)$  |  $a = 1$  setzen

$b^2 = 1 * (1 + b)$  | ausmultiplizieren

$b^2 = 1 + b$  | „ $=$ “ gegen „ $<=$ “ ersetzen

$b^2 <= 1 + b$

Anders ausgedrückt: Suche eine Zahl  $b$ , deren Quadrat  $<= b + 1$  ist:

- $b = 1$  funktioniert, denn  $1 < 1 + 1$
- $b = 2$  funktioniert nicht, denn  $4 > 2 + 1$
- $b$  muss demzufolge zwischen 1 und 2 liegen

Mit dieser Denke kann man einen PC rechnen lassen, hier ein Lösungsvorschlag in Perl:  
( Siehe nächste Seite )

```
#!/usr/bin/perl
use warnings; use strict; use bigint;

my $Sucher = 0;
my $Teiler;

foreach(1..44)
# kann beim Goldenen Schnitt auch weit über „foreach(1..100)“
# gesetzt werden
{
    $Teiler = 1 / 10**$_;
    do
    {
        $Sucher += $Teiler
    }
    while($Sucher**2 <= 1+$Sucher);
    $Sucher -= $Teiler;
}
print("Ergebnis = $Sucher \n");
```

Das Ergebnis lautet: 1.61803398874989484820458683436563811772030917

Ersetzt man:

```
while($Sucher**2 <= 1+$Sucher);
```

durch:

```
while(2**$Sucher <= 1/$Sucher);
```

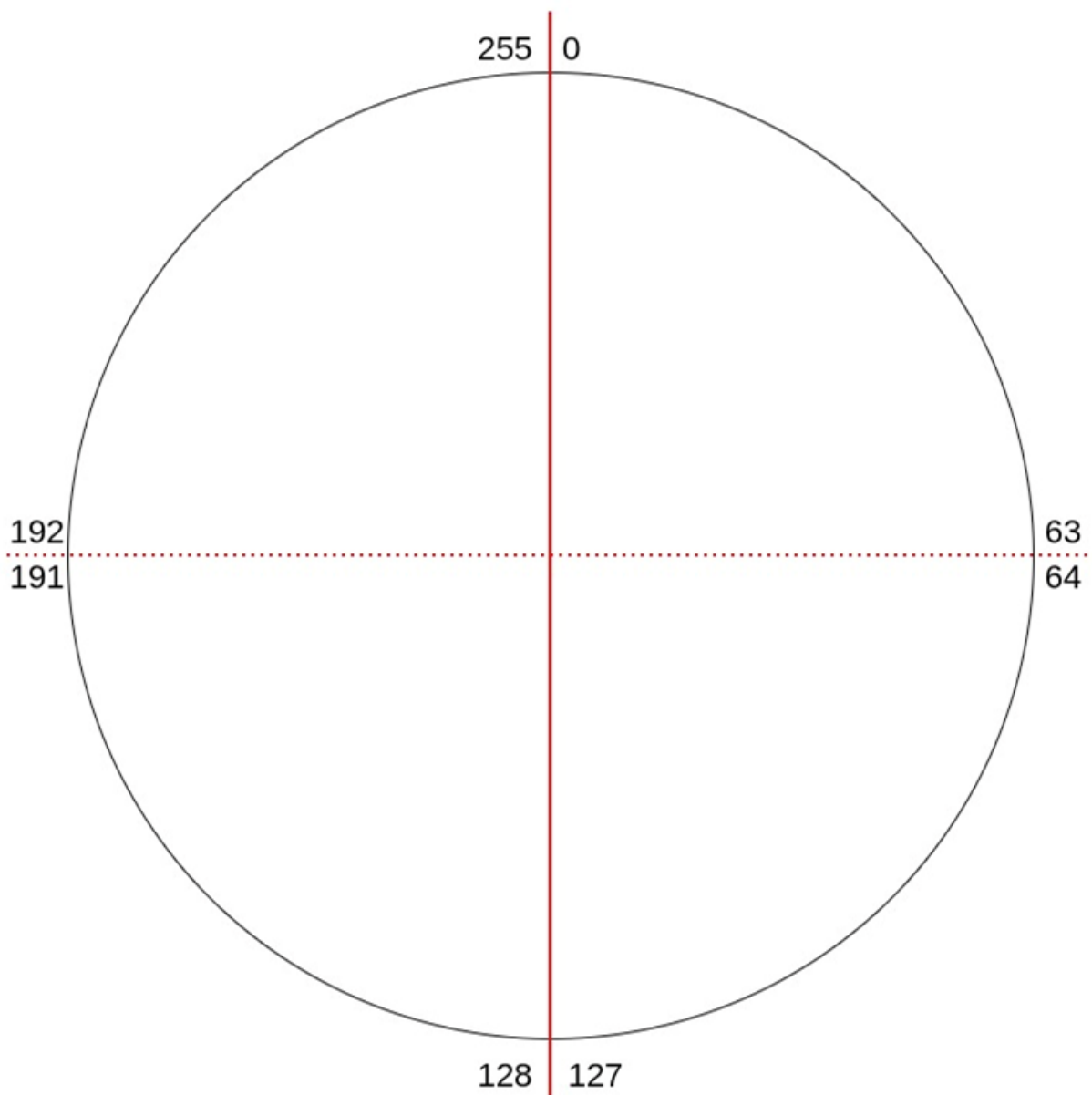
kann man sogar eines der Rätsel von Isaac Newton lösen ( $2^x = 1/x$ ).



In diesem Fall lautet das Ergebnis: 0.64118574450498598448620048211482366656297858

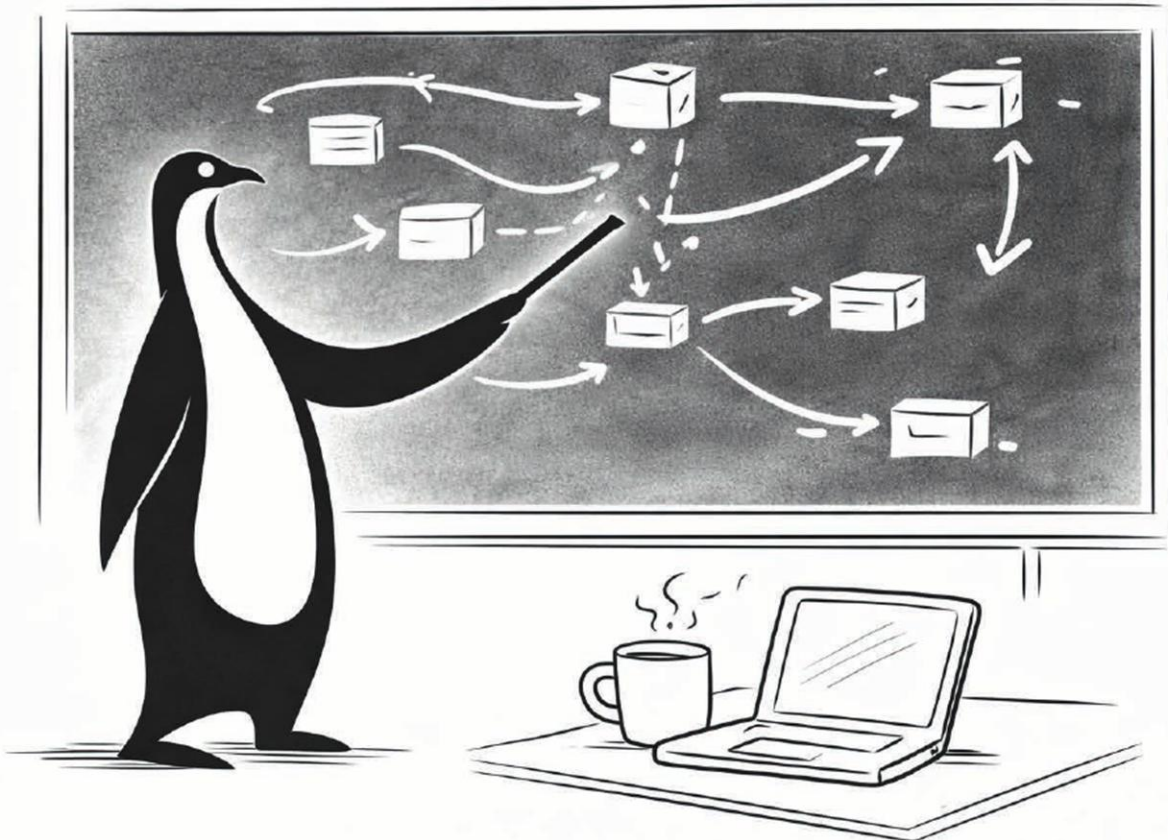
Mathematiker würden diese Lösungen vermutlich erschauern lassen, aber wie heißt es so schön:  
„Tore zählen, nicht die Elfmeter“.

## Mit Papier und Schere

### Das Mühlenbecker Wagenrad



- 1. Schnitt senkrecht  die Subnetzmaske wird um **1 (binär)** länger
- ..... 2. Schnitt wagerecht  die Subnetzmaske wird um **11 (binär)** länger



## Ein roter Faden durch das OSI-Menü

Dieses Kochbuch führt Sie Schritt für Schritt durch die Welt der Netzwerktechnik. Mit klaren Worten und verständlichen "Rezepten" erhalten Sie einen strukturierten Überblick über die wichtigsten Grundlagen der Netzwerkinfrastruktur – vom physischen Kabel bis zur Cloud.

Ob Sie neu in der Netzwerktechnik sind oder Ihr vorhandenes Wissen vertiefen möchten: Dieses Buch vermittelt Ihnen verständlich die zentralen Konzepte rund um das OSI-Modell, IP-Adressen, Router, Switches und viele weitere Netzwerkkomponenten.

Durch anschauliche Erklärungen, praxisnahe Beispiele und übersichtliche Illustrationen werden komplexe Themen greifbar und nachvollziehbar. Finden Sie Ihren eigenen roten Faden in der Welt der Bits und Bytes – und entdecken Sie, wie Netzwerke wirklich funktionieren.

*Auch wenn man den Limes nicht überwinden kann,  
so ist es doch schön ihn zu berühren.*