

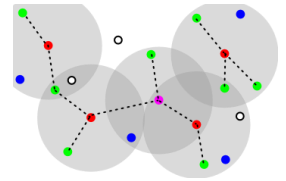
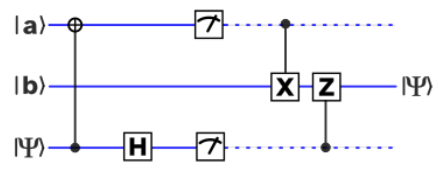
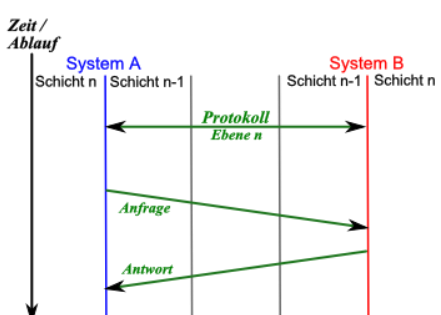
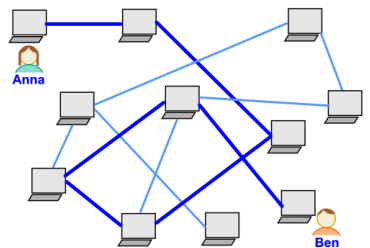
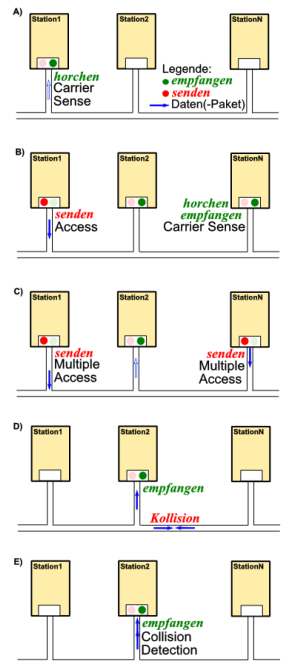
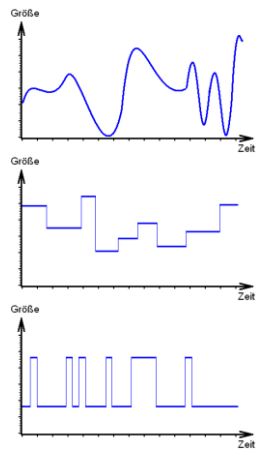
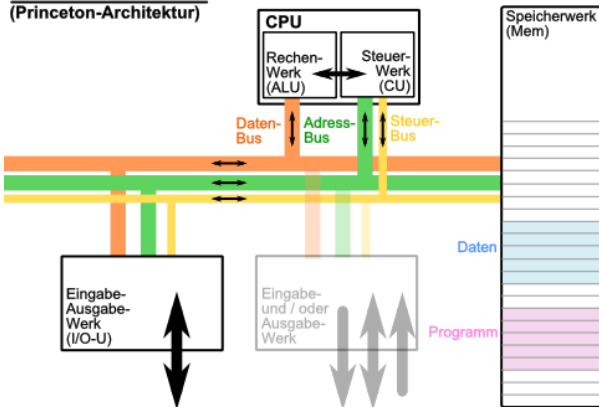
# Informatik

für die Sekundarstufe II

## - Rechner, Netzwerke und Protokolle -

Autor: L. Drews

VON-NEUMANN-Modell  
(Princeton-Architektur)



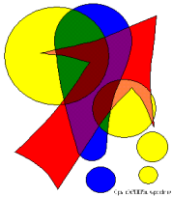
HTTP	IMAP	SMTP	...	DNS	...
TCP			UDP		
IPv4					
ARP (Address Resolution Protocol)					
Ethernet			Token-Ring		



unredigierte Version 0.12a (2023)

**Legende:**

mit diesem Symbol werden zusätzliche Hinweise, Tips und weiterführende Ideen gekennzeichnet



**Nutzungsbestimmungen / Bemerkungen zur Verwendung durch Dritte:**

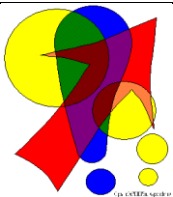
- (1) Dieses Skript (Werk) ist zur freien Nutzung in der angebotenen Form durch den Anbieter (lern-soft-projekt) bereitgestellt. Es kann unter Angabe der Quelle und / oder des Verfassers gedruckt, vervielfältigt oder in elektronischer Form veröffentlicht werden.
- (2) Das Weglassen von Abschnitten oder Teilen (z.B. Aufgaben und Lösungen) in Teildrucken ist möglich und sinnvoll (Konzentration auf die eigenen Unterrichtsziele, -inhalte und -methoden). Bei angemessen großen Auszügen gehört das vollständige Inhaltsverzeichnis und die Angabe einer Bezugsquelle für das Originalwerk zum Pflichtteil.
- (3) Ein Verkauf in jedweder Form ist ausgeschlossen. Der Aufwand für Kopierleistungen, Datenträger oder den (einfachen) Download usw. ist davon unberührt.
- (4) Änderungswünsche werden gerne entgegen genommen. Ergänzungen, Arbeitsblätter, Aufgaben und Lösungen mit eigener Autorenschaft sind möglich und werden bei konzeptioneller Passung eingearbeitet. Die Teile sind entsprechend der Autorenschaft zu kennzeichnen. Jedes Teil behält die Urheberrechte seiner Autorenschaft bei.
- (5) Zusammenstellungen, die von diesem Skript - über Zitate hinausgehende - Bestandteile enthalten, müssen verpflichtend wieder gleichwertigen Nutzungsbestimmungen unterliegen.
- (6) Diese Nutzungsbestimmungen gehören zu diesem Werk.
- (7) Der Autor behält sich das Recht vor, diese Bestimmungen zu ändern.
- (8) Andere Urheberrechte bleiben von diesen Bestimmungen unberührt.

**Rechte Anderer:**

Viele der verwendeten Bilder unterliegen verschiedensten freien Lizenzen. Nach meinen Recherchen sollten alle genutzten Bilder zu einer der nachfolgenden freien Lizenzen gehören. Unabhängig von den Vorgaben der einzelnen Lizenzen sind zu jedem extern entstandenen Objekt die Quelle, und wenn bekannt, der Autor / Rechteinhaber angegeben.

<b>public domain</b> (pd)	Zum Gemeingut erklärte Graphiken oder Fotos (u.a.). Viele der verwendeten Bilder entstammen Webseiten / Quellen US-amerikanischer Einrichtungen, die im Regierungsauftrag mit öffentlichen Mitteln finanziert wurden und darüber rechtlich (USA) zum Gemeingut wurden. Andere kreative Leistungen wurden ohne Einschränkungen von den Urhebern freigegeben.
<b>gnu free document licence</b> (GFDL; gnu fdl)	
<b>creative commons</b> (cc) 	od. neu  ... Namensnennung ... nichtkommerziell ... in der gleichen Form ... unter gleichen Bedingungen

Die meisten verwendeten Lizenzen schließen eine kommerzielle (Weiter-)Nutzung aus!



**Bemerkungen zur Rechtschreibung:**

Dieses Skript folgt nicht zwangsläufig der neuen **ODER** alten deutschen Rechtschreibung. Vielmehr wird vom Recht auf künstlerische Freiheit, der Freiheit der Sprache und von der Autokorrektur des Textverarbeitungsprogramms microsoft® WORD® Gebrauch gemacht. Für Hinweise auf echte Fehler ist der Autor immer dankbar.

---

# Inhaltsverzeichnis:

	Seite
<b>0. Einleitung .....</b>	<b>11</b>
<b>1. Rechner - Grundlagen .....</b>	<b>12</b>
<b>1.0. Grundbegriffe / Grundprinzipien .....</b>	<b>12</b>
1.0.1. alle mit EVA – oder? .....	12
1.0.2. analog oder digital – das (!) ist hier die Frage .....	13
<b>1.1. Grundlagen der digitalen Datenverarbeitung .....</b>	<b>16</b>
1.1.x. Zahlensysteme .....	16
1.1.x.y. das duale Zahlensystem .....	17
1.1.x.y. Konvertierung von Zahlen zwischen den Zahlensystemen .....	18
1.1.x.y.z. Rechnen im dualen Zahlensystem .....	21
Exkurs: Wie ging den noch die schriftliche Addition im Dezimalsystem? .....	21
1.1.x.y.z. logische Operationen im dualen Zahlensystem .....	27
Logik-Gatter – die technische Realisierung von Logik-Operationen .....	30
Umsetzung in ein Rechenwerk .....	33
1.1.x.y. das hexadezimale Zahlensystem .....	35
<b>1.2. Aufbau von Datenverarbeitungsanlagen .....</b>	<b>39</b>
Definition(en): Datenverarbeitungsanlage .....	39
Definition(en): Hardware .....	39
Definition(en): Software .....	39
Definition(en): Informatik-System / informatisches System .....	40
Definition(en): Programm .....	40
Definition(en): Information .....	40
Definition(en): Datum .....	40
Biographie: Konrad ZUSE (1910 - 1995) .....	41
1.2.1. Grundelement Speicher .....	42
1.2.2. VON-NEUMANN-Architektur .....	44
Definition(en): VON-NEUMANN-Architektur .....	45
Grobaufbau einer CPU .....	46
Exkurs: Prozessoren von morgen? .....	51
Biographie: John VON NEUMANN (1903 - 1957) .....	54
1.2.3. Prozesse .....	55
Definition(en): Prozess .....	55
Definition(en): Interrupt .....	57
Definition(en): Task .....	58
Definition(en): Thread .....	58
1.2.4. Programmierung des Rechners / der CPU .....	59
1.2.4.2. Simulation eines VON-NEUMANN-Rechner mit "Johnny" .....	60
1.2.2.2.1. Aufbau des Simulators .....	61
1.2.2.2.2. Programmierung von Johnny .....	69
1.2.2.2.2.1. Erstellen einfacher Assembler-Programme .....	69
1.2.2.2.2.2. Beobachtung des VON-NEUMANN-Befehls-Zyklus .....	77
1.2.2.2.2.3. Erstellen neuer Makro's / Assembler-Befehle .....	79
1.2.2.3. Simulation eines VON-NEUMANN-Rechner mit MOPS .....	81
Besonderheiten zum Aufbau der einzelnen Werke .....	82
Erstellen eines Assembler-Programm's .....	83
Übersicht zum Assembler-Code in MOPS (Cheat sheet) .....	84
Übersetzen des Programm's in Maschinen-Code und Simulation der Abläufe .....	87
1.2.2.4. neuartige Strukturen / Modelle / Konzepte / Erweiterungen bei VON-NEUMANN-Rechnern .....	88
1.2.5. Harvard-Architektur .....	89
Definition(en): Harvard-Architektur .....	89
1.2.6. Assoziativ-Maschine .....	90
Definition(en): Assoziativ-Architektur .....	90

1.2.7. Ternär-Rechner .....	91
1.2.7.1. Geschichtliches / Historie .....	91
1.2.7.2. Grundlagen / ternäre Logik .....	92
<b>1.3. das Schalen-Modell.....</b>	<b>96</b>
<b>1.4. Quanten-Computing.....</b>	<b>98</b>
1.4.0. Grundlagen .....	98
1.4.0.1. Historie / Geschichte.....	98
1.4.1. Qubit's .....	99
Definition(en): Qubit (Quanten-Bit) .....	100
Definition(en): Superposition .....	101
Definition(en): Verschränkung .....	101
Definition(en): Dekohärenz .....	101
Definition(en): Unterscheidbarkeit.....	101
1.4.2. Quanten-Schaltkreise – Schaltkreise für Qubit's.....	101
Definition(en): Quanten-Schaltkreis .....	102
Definition(en): Quanten-Gatter.....	103
1.4.3. Quanten-Register .....	104
Definition(en): Quanten-Register .....	105
Definition(en): Basis-Zustand.....	105
Definition(en): Amplitude .....	106
Definition(en): Superposition .....	106
1.4.4.1. Berechnen von Quanten-Registern.....	107
1.4.4.2. Veranschaulichen / Illustration von Quanten-Registern.....	109
1.4.4.3. Veranschaulichung des Messen's in Quanten-Registern .....	110
1.4.4.4. verschiedene Quanten-Gatter in Quanten-Registern .....	112
PAULI-X-Gatter, X-Gatter .....	112
PAULI-Y-Gatter, Y-Gatter .....	114
PAULI-Z-Gatter, Z-Gatter .....	114
HADAMARD-Gatter, H-Gatter .....	117
Anwendung des HADAMARD-Gatter's zur Erzeugung von echtem Zufall .....	118
Nutzung des HADAMARD-Gatter's in Quanten-Registern .....	119
controlled NOT-Gatter, CNOT-Gatter .....	121
TOFFOLI-Gatter .....	123
weitere Objekte in Quanten-Schaltkreisen.....	123
1.4.4. Quanten-Gatter zu Quanten-Schaltkreisen kombiniert.....	125
1.4.5. Quanten-Algorithmen .....	128
1.4.6. Quanten-Computer.....	129
Definition(en): Topologie.....	130
<b>1.5. Virtualisierung.....</b>	<b>131</b>
Virtualisierung / Simulation von Rechnern / Servern.....	131
Einbau einer Zwischen-Schicht .....	132
Emulatoren.....	132
Java Runtime Environment (JRE).....	132
.Net .....	132
Wine.....	132
echte Virtualisierung von Rechner-Systemen.....	133
virtualBox .....	133
VMware.....	133
Docker.....	133
Cloud-Computing .....	133
<b>1.6. Kenndaten für Computer-Systeme .....</b>	<b>135</b>
<b>2. Netzwerke - Grundlagen .....</b>	<b>136</b>
<b>2.1. Grundlagen Netzwerke .....</b>	<b>136</b>
Definition(en): Netzwerk .....	136
Topologien .....	137
Topologie: Struktur-Aspekt .....	137



Ring-Topologie .....	137
Bus-Topologie .....	138
Stern-Topologie.....	138
Maschen-Netz als offene Misch-Topologie .....	139
weitere Topologien .....	141
Topologie: Ausdehnungs-Aspekt.....	142
GAN .....	142
Definition(en): GAN .....	142
WAN.....	142
Definition(en): WAN.....	142
MAN .....	143
Definition(en): MAN – Metro Area Network.....	143
LAN .....	144
Definition(en): LAN – Local Area Network .....	144
PAN.....	144
Definition(en): PAN – Personal Area Network.....	144
Ad-hoc- und Pico-Netzwerke .....	145
Topologie: (inhaltlicher) Abgrenzungs-Aspekt .....	147
Intranet.....	147
Definition(en): Intranet .....	147
Internet.....	148
Definition(en): Internet .....	148
Kommunikations-Konzepte .....	149
Client-Server-Konzept .....	149
Peer-to-Peer-Konzept .....	150
<b>2.2. Grundlagen Datenübertragung .....</b>	<b>151</b>
2.2.x. allgemeines Modell der Kommunikation .....	151
2.2.x. Medien für die Daten-Übertragung .....	152
2.2.x.y. Kabel .....	152
2.2.x.y.z. Luftkabel .....	152
2.2.x.y.z. Verlegekabel .....	153
2.2.x.y.z. Stromleitungen .....	153
2.2.x.y.z. Erdkabel .....	153
2.2.x.y.z. Unterseekabel .....	153
2.2.x.y. Lichtwellenleiter .....	154
2.2.x.y.z. Single-Mode .....	154
2.2.x.y.z. Multi-Mode .....	154
2.2.x.y. Funkwellen.....	155
2.2.x.y.z. Infrarot.....	155
2.2.x.y.z. Richtfunk .....	155
2.2.x.y.z. Satellitenfunk.....	155
2.2.x.y.z. Landfunk .....	155
2.2.x.y.z. Lokalfunk.....	155
2.3.x. Modulations-Verfahren .....	156
2.3.x.y. Amplituden-Modulation .....	156
2.3.x.y. Frequenz-Modulation .....	157
2.3.x.y. Phasen-Modulation .....	158
2.4.x. Schicht-Modelle .....	159
2.4.x.y. ISO-OSI-Schichtmodell.....	159
Überblick über das OSI-Modell und Einordnung des TCP/IP-Modell's .....	161
Definition(en): ISO-OSI-Modell .....	162
Vereinfachtes Modell für Paketierung im Internet – Waren-Ver- und Entpackung.....	168
2.4.x.y. DoD-Modell.....	169
2.4.x.y. TCP/IP-Referenz-Modell .....	170
IP-Schicht.....	171
Exkurs: MAC-Adresse .....	172
IP-Adresse .....	173

Berechnungen der Netzwerk-Adressen-Teile: .....	181
Berechnungen der optimalen Netz- / Subnetz-Maske: .....	182
Netzwerk-Automatisierung .....	190
DHCPv6 .....	191
Internet-Ressourcen-Adressen .....	193
OSI-Adressen .....	194
Transport-Schicht .....	195
<b>3. Protokolle - Grundlagen .....</b>	<b>198</b>
Definition(en): Protokolle .....	199
<b>3.x. Kommunikations-Modi .....</b>	<b>201</b>
3.x.1. synchrone Datenübertragung .....	201
3.x.2. asynchrone Datenübertragung .....	201
Verbindungs-orientierte Daten-Übertragung .....	202
Definition(en): Verbindungs-orientierte Kommunikation .....	203
Verbindungs-lose Daten-Übertragung .....	205
Definition(en): Verbindungs-lose Kommunikation .....	205
CSMA/CD-Zugriffs-Verfahren .....	206
Token-Ring-Verfahren .....	208
ALOHA-Verfahren .....	211
<b>Netzwerk-Protokolle .....</b>	<b>212</b>
Address Resolution Protocol - ARP .....	212
Internet Message Control Protokoll - ICMP .....	213
Dynamic Host Configuration Protokoll - DHCP .....	217
DNS-Service .....	218
http- und https-Protokoll .....	221
<b>Codierung .....</b>	<b>224</b>
Definition(en): Codierung .....	224
Codierung von 0 und 1 auf Schicht 1 .....	224
Manchester-Kodierung .....	224
4B/5B-Kodierung .....	225
8B/10B-Kodierung .....	226
weitere Kodierungen .....	227
MLT-3-Kodierung .....	227
PAM5- und Trellis-Kodierung .....	227
RC5-Code .....	228
Umsetzung einer (möglichen) Decodierung des RC5-Codes .....	230
Decabit-Impulsraster .....	230
Unified Diagnostic Services (UDS) .....	232
<b>Chiffrierung .....</b>	<b>233</b>
Definition(en): Chiffrierung .....	233
<b>4. praktische Netzwerke und ihre Protokolle .....</b>	<b>234</b>
<b>4.1. das Ethernet .....</b>	<b>234</b>
Definition(en): Ethernet .....	234
Standard-Ethernet .....	235
10Base5 .....	235
10Base2 .....	235
10BaseT .....	235
10BaseF .....	235
Fast-Ethernet .....	236
Gigabit-Ethernet .....	236
zukünftige Ethernets .....	237
Aufbau eines Ethernet-Paketes (mit maximalen IPv4- / TCP-Daten) .....	238
4.1.x. Ethernet-Geräte .....	239
2.5.x.y. Repeater, Hub's, Switches, Router, Brigdes, Gateway's .....	239
2.5.x.y.z. Repeater .....	241
Definition(en): Repeater .....	241

2.5.x.y.z. Hub .....	242
Definition(en): Hub .....	242
2.5.x.y.z. Bridge.....	243
Definition(en): Bridge.....	243
2.5.x.y.z. Switch .....	244
Definition(en): Switch.....	245
2.5.x.y.z. Router .....	247
Definition(en): Router .....	247
2.5.x.y.z. Gateway .....	248
Definition(en): Gateway .....	248
Wege-Wahl-Verfahren.....	249
Domain Name Service (DNS).....	249
Datenübertragung .....	249
der heimische WLAN-Router – ein kleines Universal-Gerät.....	253
<b>4.2. Simulation von Netzen mit Filius .....</b>	<b>256</b>
4.2.0. Wege zu Filius.....	256
4.2.1. Aufbau von Netzen.....	257
4.2.1.1. Einrichten und Nutzen von Netzendgeräten in Filius.....	258
4.2.1.2. Verbinden von Netzendgeräten in Filius .....	262
4.2.1.3. Clients und Server .....	269
4.2.2. Verbindung von Netzen.....	272
<b>4.3. Simulation von Netzen mit NetEmul .....</b>	<b>274</b>
4.3.0. Einführung.....	274
4.3.1. Programm-Start und Aufbau eines Netzwerkes.....	274
4.3.2. Simulationen und Beobachtungen im Netzwerk .....	277
kleiner Hilfs-Algorithmus zum Erstellen von Netzwerk's- und Kommunikations- Protokollen .....	278
4.3.2.x. ARP-Nachrichten .....	281
4.3.2.x. IP-Nachrichten .....	282
4.3.2.x. zusätzliche Programme auf den Netzgeräten .....	282
4.3.3. Erweiterung um weitere Netzwerk-Komponenten und -Verbindungen .....	284
4.3.3.1. Einbau eines Server's zum Bereitstellen eines Dienstes.....	284
4.3.3.2. Einbau eines Router's zum Verbindung von unterschiedlichen Netzen.....	287
4.3.3.3. Einbau einer Bridge zum Verbindung von Teil-Netzwerk .....	287
4.3.3.4. Verwendung eines Gateway's – zentrale Kontrolle .....	288
4.3.3.5. Verwendung eines Hub's in Netzen .....	288
<b>4.x. spezielle Netze und Protokolle.....</b>	<b>291</b>
<b>ISDN .....</b>	<b>291</b>
<b>DSL .....</b>	<b>291</b>
<b>Kabelmodem .....</b>	<b>291</b>
<b>WLAN.....</b>	<b>292</b>
<b>GSM .....</b>	<b>292</b>
<b>UMTS .....</b>	<b>292</b>
<b>Internet .....</b>	<b>293</b>
Biographie: Tim BERNERS-LEE (1955 - ).....	295
<b>TCP/IP .....</b>	<b>296</b>
IPv4.....	296
IPv6.....	296
IPv6 und das IoT .....	299
UDP .....	300
TCP.....	300
Übertragungsrahmen Ethernet .....	301
wichtige Internet-Protokolle .....	302
electronic Mail (eMail) .....	302
POP / IMAP .....	302
SMTP .....	302
Hypertext-System.....	302

Hypertext Transport Protocol (http, HTTP).....	302
Datei-Übertragungs-Protokolle .....	303
File Transfer Protocol (FTP) .....	303
das Usenet .....	304
<b>4.x. Clouds – Arbeiten in der Wolke.....</b>	<b>305</b>
<b>4.x. IoT – Internet of Things.....</b>	<b>306</b>
4.x.y. MQTT – das Protokoll für IoT .....	307
4.x.y.z. alternative Prokollle für IoT-Anwendungen.....	312
http als IoT-Protokoll.....	312
CoAP .....	312
XMPP .....	313
iBeacon .....	313
<b>4.x. IoV – Internet of Value / Blockchain-Technologie .....</b>	<b>314</b>
Problem-Fragen für Selbstorganisiertes Lernen .....	314
Definition(en): Blockchain .....	314
Sicherheit in Blockchain-Systemen.....	322
4.x.y. Krypto-Zahlungssysteme .....	324
4.x.y.0. Grundlagen .....	324
4.x.y.1. Anwendungs-Beispiele zur Blockchain-Technologie .....	327
4.x.y.2. Blockchain-Hauptanwendung: Finanzwesen und Krypto-Währungen .....	333
4.x.y.2.0. "normale" Bankgeschäfte (ohne Blockchain) .....	333
4.x.y.2.1. Bankgeschäfte mit Blockchain.....	333
Definition(en): Bitcoin .....	334
4.x.y.3. Anwendung: Cloud's und Cloud-Computing.....	337
4.x.y.4. Anwendung: Internet of Things (IoT) .....	338
4.x.y.4. Anwendungs-Bereich: Energie-Sektor .....	340
4.x.y.5. Anwendungs-Bereich: Logistik .....	341
4.x.y.6. Anwendungs-Bereich: Identitäts-Management.....	342
4.x.y. Blockchain selber programmieren.....	344
4.x.y.0. Grundlagen / Ausgangspunkt.....	344
4.x.y.1. einfache Transaktions-Historie .....	344
4.x.y.2. Manipulations-Möglichkeiten der Transaktions-Historie.....	344
4.x.y.3. Absicherung der Transaktions-Historie.....	344
4.x.y.4. ein Blockchain-System a'la Bitchoin .....	345
4.x.y.4.1. ein Blockchain-System a'la Bitchoin in Python .....	345
4.x.y.4.2. ein Blockchain-System a'la Bitchoin in JAVA.....	348
<b>5. Datenschutz und Datensicherheit.....</b>	<b>354</b>
<b>6. Sicherheit in Datennetzen.....</b>	<b>355</b>
<b>7. Netzwerk-Virtualisierung .....</b>	<b>357</b>
<b>7.1. virtuelle LAN's - VLAN.....</b>	<b>358</b>
<b>7.2. Port-Channels.....</b>	<b>359</b>
<b>7.3. Tunnel .....</b>	<b>360</b>
<b>7.4. Virtuelle private Netzwerke.....</b>	<b>362</b>
<b>7.5. Linux virtueller Switch .....</b>	<b>364</b>
<b>7.6. Linux als Router .....</b>	<b>366</b>
<b>7.6. VXLAN.....</b>	<b>367</b>
<b>7.6. Virtual Routing and Forwarding - VRF.....</b>	<b>368</b>
VRF mittels MPLS .....	369
VRF zum Managen .....	369
<b>7.8. KVM (Kernel Virtual Machine) .....</b>	<b>370</b>
Tunnel.....	370
<b>7.8. VMware .....</b>	<b>371</b>
7.8.x. VMware NSX .....	372
<b>7.9. Software Defined WAN - SD-WAN.....</b>	<b>373</b>
7.9.1. Locator ID Separation Protocol - LISP .....	375

7.9.1.1. Proxy-xTR .....	377
7.1.1.2. Bewertung von LISP .....	377
7.9.2. Viptela .....	377
7.9.2.1. Zero Touch Provisioning .....	379
7.9.2.2. ausgewählte Feature's von SD-WAN und zugehörigen Lösungen .....	379
<b>8. Netzwerke und Protokolle am Beispiel "Internet" .....</b>	<b>380</b>
<b>8.0. Einleitung .....</b>	<b>380</b>
8.0.x. Digitalisierung .....	380
<b>8.1. kleine Geschichte des Internet's .....</b>	<b>382</b>
8.1.0. Kommunikation vor dem Internet .....	382
8.1.1. Computer als Voraussetzung für moderne Kommunikation .....	384
Geschichte der Computer-Technik .....	384
8.1.2. Entstehung des Internet's .....	387
8.1.3. Smartphone's als neue Dimension der Internet-Nutzung .....	392
<b>8.2. Rechnernetze als Basis des Internet's .....</b>	<b>394</b>
Bit's und Byte's .....	394
binäre, dezimale und hexadezimale Zahlen-Darstellung .....	395
Netzwerke und Netzwerk-Typen .....	395
Wer ist der Chef im Netzwerk? .....	399
Lokale Netzwerke (LAN / WLAN) .....	400
Kopplung von LAN's .....	401
Ethernet .....	402
WLAN – lokale Funk-Netzwerke .....	404
Sicherheit in Funk-Netzen .....	405
Netze für größere Entfernungen – WAN's .....	406
Routing im WAN .....	407
Cloud's und Cloud-Computing .....	408
<b>8.3. Internet – das Netz der Netze .....</b>	<b>409</b>
8.3.1. Kopplung der Vielzahl von Netzwerken - Internetworking .....	409
Zugänge zum Internet .....	413
8.3.2. Protokolle der Vermittlungsschicht - das Internet-Protokoll IP .....	415
8.3.2.1. Internet-Protokoll Version 4 (IPv4) .....	418
Routing bei IPv4 .....	419
Exkurs: DIJKSTRA-Algorithmus .....	421
8.3.2.2. Internet-Protokoll Version 6 (IPv6) .....	422
8.3.3. Protokolle der Transport-Schicht .....	423
8.3.3.1. TCP – Transmission Control Protocol .....	424
Aufbau eines TCP-Paket's .....	425
Verbindungs-Aufbau für eine TCP-Nachrichten-Übertragung .....	427
Ablauf / Verlaufs-Protokoll einer TCP-Nachrichten-Übertragung .....	428
Retransmission (Neuübertragung von fehlenden oder fehlerhaften Datenpaketen) .....	429
Abbau einer TCP-Nachrichten-Übertragung .....	430
Fluß- und Überlast-Kontrolle im TCP .....	431
TCP-Verwaltung als Endlicher Automat (EA) .....	433
TCP-Port's .....	434
Absicherung des Daten-Transport's auf der Transport-Schicht .....	435
8.3.3.2. UDP – User Datagram Protocol .....	437
8.4.1. grundlegende Protokolle .....	440
ICMP – Internet Control Message Protocol (RFC 792 / 1256) .....	440
ARP – Adress Resolution Protocol .....	442
SNMP – Simple Network Management Protocol .....	442
DHCP – Dynamic Host Configuration Protocol .....	443
NFS – Network File System .....	443
TCP-Beobachtung mit Wireshark .....	444
<b>8.4. Internet-Anwendungen .....</b>	<b>446</b>
DNS – Domain Name Service .....	446

electronic Mail – eMail – POP / SMTP / IMAP .....	450
File Transport Protocol - FTP .....	454
Instant Messaging / Chat - .....	454
Hypertext Transfer Protocol - http / www .....	454
Exkurs: Aufbau einer HTML-Datei .....	457
Suchmaschinen .....	458
soziale Netzwerke .....	458
8.4.x. erweiterte Protokolle .....	459
Media-Streaming .....	459
Mediatheken, aNetflix und Co .....	460
Voice over IP – VoIP .....	460
online-Gaming .....	461
online-Banking .....	461
Arbeiten in der Cloud .....	462
8.4.x. IoT – Internet of Things (Internet der Dinge) .....	463
smart Home / Gebäude-Steuerung .....	464
8.4.x. Anwendungen / Protokolle der nahen Zukunft .....	464
<b>8.5. Internet-Sicherheit .....</b>	<b>465</b>
8.5.x. Sicherheits-Ziele und Angriffs-Szenarien .....	467
online-Banking .....	470
Probleme beim online-Banking .....	470
social Networking .....	470
Probleme beim social Networking .....	471
Definition(en): Schwachstelle .....	471
Definition(en): Exploit .....	471
Definition(en): Schadcode .....	471
Definition(en): Sicherheitsvorfall .....	472
Definition(en): Malware .....	473
Beispiele für Angriffs-Szenarien .....	475
Phishing .....	476
WannaCry-Kampagne .....	477
Zero-Day-Angriff .....	478
8.5.x. digitale Identität .....	479
Definition(en): digitale Identität .....	479
Definition(en): Authentifizierung .....	480
Definition(en): Autorisation .....	480
8.5.x. Verschlüsselung im Internet .....	481
Definition(en): Kryptographie .....	481
Definition(en): Kryptologie .....	483
Definition(en): Steganographie .....	485
8.5.x. allgemeine Sicherheits-Empfehlungen .....	486
sicheres Passwort .....	486
2-Faktor-Authentifizierung .....	486
Datenträger-Verschlüsselung .....	486
Prinzip der Daten-Sparsamkeit .....	487
Updates bei Programmen .....	487
Updates für das Betriebssystem .....	488
Anti-Viren-Software / Internet-Security-Suiten .....	488
Backup's / Datensicherungen .....	489
Firewall .....	489
<b>Literatur und Quellen: .....</b>	<b>490</b>

---

# 0. Einleitung

Dieses Skript besteht aus mehreren thematisch-didaktisch orientierten Bereichen.

## **allgemeine Grundlagen (Kapitel 1 bis 7)**

Die Kapitel 1 und 2 beschäftigen sich mit den Grundlagen von Rechnern und Netzwerken. Sie sind Voraussetzungen zum Verständnis moderner Rechner und Netzwerke.

Die verwendeten Arbeitsweisen und Regeln – die sogenannten Protokolle – werden dann im Kapitel 3 besprochen.

Mit den praktischen Netzen und der Simulation solcher Netze beschäftigt sich Kapitel 4. Hier stellen wir u.a. zwei Simulations-Programme ("Filius" und "NetEmul") vor.

Die Kapitel 4 und 5 sind als theoretische Einheit zu betrachten. Hier geht es um Datensicherheit und Daten-Schutz. Diese Kapitel sind weitestgehend unabhängig von den anderen Kapiteln. Die Grundlagen werden aber für ein tiefgreifendes Verständnis gebraucht.

## **spezielle Netze (Kapitel 7)**

hier gehen wir weit über den klassischen Unterricht hinaus. Das Kapitel bietet einige Aspekte moderner Netze, die heute zur gängigen Praxis gehören. Interessierte können ja zumindestens mal rüberlesen.

Weiter hinten im **Projekt-orientierten Abschnitt (Kapitel 8)** wird die Vernetzung aus der Sicht des Internet's aufgezeigt.

Insgesamt sind durch die verschiedenen Nutzungs- und Herangehens-Weisen Inhalte doppelt. Wiederholungen schaden im Allgemeinen auch nicht. Wer fachlich sicher ist, kann einzelne Absätze oder Seiten überfliegend lesen oder ganz auslassen. Wenn was unklar bleibt, dann kann man eine der doppelten / mehrfachen Abhandlungen durcharbeiten.

# 1. Rechner - Grundlagen

Der Computer arbeitet deshalb so schnell, weil er nicht denkt.  
Daniel LAUB

## 1.0. Grundbegriffe / Grundprinzipien

### 1.0.1. alle mit EVA – oder?

Eingabe	Verarbeitung		Ausgabe
<b>Eingabeeinheit</b> Daten werden vom Peripheriegerät eingegeben	<b>Rechenwerk</b> führt zur Verarbeitung der Daten arithmetische und logische Operationen durch	<b>Steuereinheit</b> steuert den Datenverkehr zwischen Registern und Speichern, sowie den Programmablauf	<b>Ausgabereinheit</b> Daten werden vom Peripheriegerät ausgegeben
<b>Speicher</b> speichert Programme, Daten und Befehle			

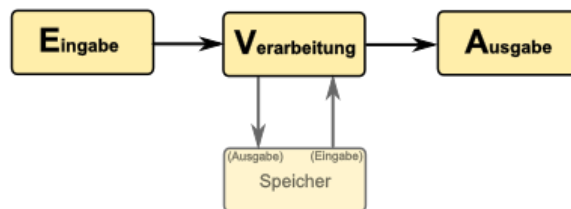
Auch wenn der EVA-Durchgang die entscheidende Struktur ist, so wird man heute immer mehr bewußt, wie wichtig der Speicher eigentlich ist.

In der Literatur wird deshalb auch immer häufiger vom EVAS-Prinzip bzw. EVAS-Modell gesprochen.

Problematisch ist hier aber die Reihenfolge der Buchstaben.

Da wären Abkürzungen EVSA oder ESVA besser geeignet – die klingen aber gar nicht gut. Um die Sonderstellung des Speichers zu betonen, findet man auch die Benennung als EVA(S)-Prinzip.

Für die Programmierung spielt die Speicherung eine nebenläufige Rolle. Es wird ständig mit dem Speicher hantiert. Eingaben werden dort abgelegt. Bei der Verarbeitung wird lesend und schreibend auf den Speicher zugegriffen. Für die Ausgabe werden dann schließlich die Daten wieder aus dem Speicher herausgeholt und angezeigt.

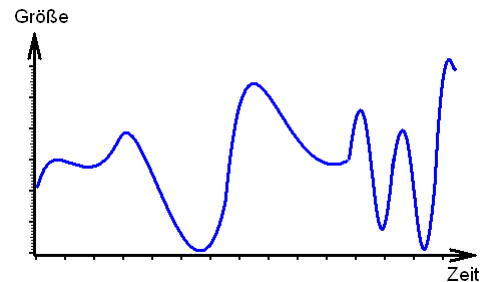




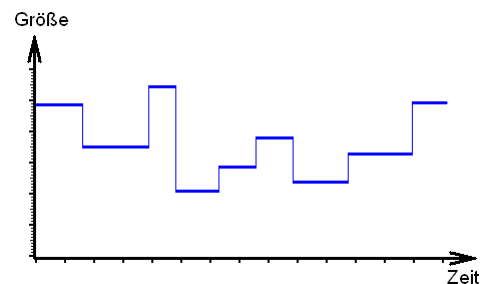
## 1.0.2. analog oder digital – das (!) ist hier die Frage

exakt müsste es analog oder diskret als Gegenüberstellung heißen

unter analogen Signalen verstehen wir solche, die innerhalb eines bestimmten Bereiches unendlich viele Werte (Zustände) einnehmen können die Werte sind stufenlos, kontinuierlich

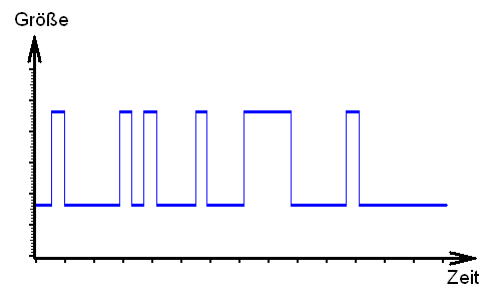


diskrete Signale besitzen innerhalb eines bestimmten Bereiches eine abzählbare Menge von Zuständen; Werte haben / stehen für bestimmte Stufen



sind es zwei verschiedene Zustände dann nennen wir das dual, digital bzw. binär

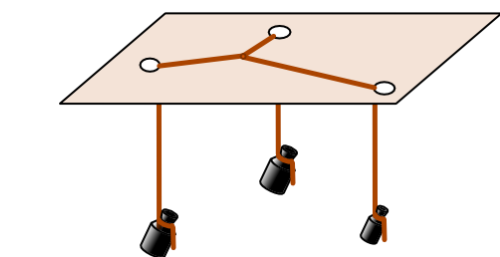
digital kommt ursprünglich von lat.: digital = Finger und meinte mit den Fingern abzählbar



analoge Lösungen für viele Optimierungs-Aufgaben bekannt (z.B. effektivstes Rohrnetz / bester Standort für eine Wasserversorgung usw.) praktisch ohne Zeitaufwand für die "Berechnung"

Addition und Subtraktion einfach realisierbar  
 $2\text{ V} + 3\text{ V} = 5\text{ V}$   
 $6\text{ V} - 2\text{ V} = 4\text{ V}$

Nachteil immer nur bestimmte Probleme lösbar



analoge Lösungsmöglichkeit für die Optimierung eines Rohrsystems (die Gewichte modellieren den Verbrauch an den Standorten (Löcher in der Platte))

In der Frühzeit der elektronischen Datenverarbeitung gab es viele Umsetzungen von Analog-Rechner (z.B. Addition und / oder Subtraktion von Spannungen)

---

Entscheidung für digitales System, wegen der Einfachheit und Universalität der umzusetzenden Verfahren

Lösung sehr unterschiedlicher Probleme möglich

Nachteil: es müssen Programme zur Lösung jeder Problemklasse geschrieben werden (Algorithmenentwurf)

Programmablauf benötigt Zeit

praktisch basieren digitale Datenverarbeitungsanlagen auf zwei Zuständen und drei Arbeitsverfahren, die elektrisch / elektronisch umgesetzt werden müssen:

es werden zwei Zustände definiert, z.B. zwei Spannungen (z.B. 0 V und 5V)

z.B. als Zustand0 und Zustand1 bezeichnet, werden den informatischen Zuständen (Bits) 0 und 1 entsprechend verwendet

### ***Grundlegende Arbeitsverfahren (Operationen) in digitalen Systemen***

- **SETZE Zustand1**
- **WECHSEL\_ZUSTAND**
- **VERGLEICHE (mit Zustand1)**

grundlegend sind diese Operation für ein einzelnes Bit

alle anderen Leistungen (z.B. für ein Byte oder das Verrechnen von Bits oder Bytes) werden aus diesen Operationen zusammengesetzt

natürlich werden in der Praxis weitere – ev. auch komplexere – Operationen in die Hardware eingebaut

Verweis auf endliche Automaten / TURING-Maschinen in einem anderen Skript

erste – verbreitete – Rechner waren für 4 bit ausgelegt,  
Initialzündung und besonders erfolgreich 8 bit-Generation  
dann 16 und 32 bit  
derzeit (2015) typisch 64 bit

### **Aufgaben:**

- 1.
- 2.
- 3.

Ternär-Computer Setun (1956)

---

### Aufgaben:

**1. Bewerten Sie die folgenden Ausgaben / Geräte als analoge, diskrete und / oder digitale (duale) Signale!**

- |                                                   |                                     |
|---------------------------------------------------|-------------------------------------|
| a) Uhr mit Ziffern-Anzeige                        | b) Schallplatte                     |
| c) Digital-Foto                                   | d) automatische Treppen-Beleuchtung |
| e) CD                                             | f) gesprochener Text                |
| g) Alarmanlage                                    | h) Musik aus Verstärker             |
| i) klassische Zeiger-Uhr mit mechanischem Uhrwerk |                                     |
| j) Ein/Aus-Schaltung bei einem Radio-Gerät        |                                     |
| k) Text in Zeitung                                | l) Foto (klassische Filmkamera)     |
| m)                                                | n) modernes TV-Gerät                |

**2. Suchen Sie sich aus Ihrer Lebenswelt 10 (technische) System aus und ordnen Sie diese den analogen, diskreten und / oder digitalen (dualen) Systemen zu!**

**3. Ein älterer Informatiker behauptet: "Eigentlich wäre es in den Frühzeiten der Datenverarbeitungs-Technik cleverer gewesen auf ein trinäres System zu setzen. Dann hätte man zwei Signal-Stufen für 0 und 1 sowie eine Signal-Stufe (z.B. 0 V) gehabt. Mit der Null-Stufe hätte man besser zwischen Nichtarbeit und der Daten-0 unterscheiden können."**

**4. Woran könnte es gelegen haben, dass man sich für ein binäres Datenverarbeitungs-Modell entschieden hat?**

**5. Überlegen Sie sich, ob man auch mit anderen (vielleicht weniger) digitalen Grundoperationen auskommen könnte?**

---

## 1.1. Grundlagen der digitalen Datenverarbeitung

### 1.1.x. Zahlensysteme

#### Aufgaben:

- 1. Schreiben Sie ein Programm (in der bevorzugten Programmiersprache) mit dem Sie prüfen wieviele Zeichen (Symbole) und wieviele Stellen man benötigt! Prüfen Sie die Stellen-System mit 2 bis 100 Zeichen? Lassen Sie sich die Zahl auch immer im entsprechenden System anzeigen! Benutzen Sie als Symbol-Tabelle die ASCII-Zeichen von Nr. 48 ("0") bis 57 ("9"), 65 ("A") bis 90 ("Z"), 97 ("a") bis 122 ("z") und 128 bis 163!*
- 2. Analysieren Sie die Anzahl der notwendigen Symbole (von 2 bis 100), die notwendige Stellen-Anzahl sowie die Kosten für Speicherung (Produkt aus Stellen-Anzahl und der notwendigen Stellen-Anzahl) für die folgenden Zahlen:*

a) 10	b) 20	c) 50	d) 100	e) 200	f) 500
d) 1'000	e) 10'000	f) 100'000	g) 1'000'000	h) 10'000'000	
- 3. Ist das Dual-System das kostengünstigste Zahlen-Darstellungssystem? Begründen Sie Ihre Meinung!*

## 1.1.x.y. das duale Zahlensystem

Pos.	Pot.	dual (Nibble)	dez.
0	$2^0$	0001	1
1	$2^1$	0010	2
2	$2^2$	0100	4
3	$2^3$	0000 1000	8
4	$2^4$	0001 0000	16
5	$2^5$	0010 0000	32
6	$2^6$	0100 0000	64
7	$2^7$	0000 1000 0000	128
8	$2^8$	0001 0000 0000	256
9	$2^9$	0010 0000 0000	512
10	$2^{10}$	0100 0000 0000	1'024
11	$2^{11}$	0000 1000 0000 0000	2'048
12	$2^{12}$	0001 0000 0000 0000	4'092
13	$2^{13}$	0010 0000 0000 0000	8'192
14	$2^{14}$	0100 0000 0000 0000	16'384
15	$2^{15}$	0000 1000 0000 0000 0000	32'768
16	$2^{16}$	0001 0000 0000 0000 0000	65'536
17	$2^{17}$	0010 0000 0000 0000 0000	131'072
18	$2^{18}$	0100 0000 0000 0000 0000	262'144
19	$2^{19}$	0000 1000 0000 0000 0000 0000	524'288
20	$2^{20}$	0001 0000 0000 0000 0000 0000	1'048'576

### Aufgaben:

- 1.
- 2.
- 3.

## 1.1.x.y. Konvertierung von Zahlen zwischen den Zahlensystemen

$$\text{zahl} = \sum_{i=0}^n a_i \cdot b^i = a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0$$

a ... Ziffer an der Stellenposition ; b ... Basis des Zahlensystems

(Achtung! wegen eines Bugs im Formeleditor von Word sind die Über- und Unterschrift beim Summen-Symbol falsch neben dem Symbol dargestellt!)

### Aufgaben:

- 1.
2. *Der Informatik-Professor lädt seine Studenten nach einer Vorlesung über Dualzahlen zu einem Orangensaft anlässlich seines 1000000. Geburtstag ein. Zu welchem Geburtstag sollten die Studenten wirklich gratulieren? Erlebt der Professor eigentlich den 10000000. Geburtstag?*
- 3.

### Umwandlung einer Dezimal-Zahl in eine hexadezimale

Die dezimal Zahl wird ganzzahlig durch 16 (- die neue Basis -) geteilt. Der Teiler wird im nächsten Schritt zum Teilen weiterverwendet und der Rest in das passende Symbol umgewandelt.

		neue Basis					Symbol
14972	:	16	=	935	Rest	12	→ <b>C</b>
935		16		58	Rest	7	<b>7</b>
58		16		3	Rest	10	<b>A</b>
3		16		0	Rest	3	<b>3</b>
							<b>3A7C</b>

Ist der Rest Null, dann ist man fertig.

Die Ergebnis-Symbol-Kette beginnt mit dem letzten ermittelten Symbol (ev. noch eine Null voranstellen) und endet mit dem ersten ermittelten Symbol.

### Aufgaben:

1. *Ermitteln Sie die passenden Hexadezimalzahlen zu den folgenden Dezimal-Zahlen!*
2. *Bestimmen Sie die fehlenden Darstellungen jeweils gleicher Zahlenwerte! (das rote Zahlensystem und Beispiel ist für Fortgeschrittene!)*

System	Bsp.	1	2	3	4	5	6	7
dual					010111101			
oktal				0573				
dezimal	10							
hexadezimal			02E					
vigesimal							0G6A	
sexagesimal								0Z1

---

3.

---

### **Algorithmus zum Umrechnen einer Dezimal-Zahl in eine Zahl mit bestimmter Basis**

Festlegen / Eingeben der Basis  $\rightarrow$  bas

Festlegen der Symbol-Liste  $\rightarrow$  sym[0 ... bas-1]

Eingeben der (Dezimal)-Zahl  $\rightarrow$  dez

Leeren der Ergebnis-Liste  $\rightarrow$  erg[]

Wiederholen solange bis

! Suchen des nächst höheren Potenz-Wertes  $\rightarrow$  für führende Null

pot = 0

potwert = basis<sup>pot</sup>

Wiederhole solange bis potwert > dez

    Erhöhe pot           !pot = pot + 1

    potwert = basis<sup>pot</sup>

dez  $\rightarrow$  rest

Wiederhole solange pot >= 0

    potwert = basis<sup>pot</sup>

    Teile rest ganzzahlig durch potwert  $\rightarrow$  symwert + rest

    Wähle Symbol  $\rightarrow$  sym[symbolwert] und hänge an Ergebnis-Liste an  $\rightarrow$  erg = erg + sym

Ausgeben erg

### **Algorithmus zum Umrechnen einer Zahl mit bestimmter Basis in eine Dezimal-Zahl**

Festlegen / Eingeben der Basis  $\rightarrow$  bas

Eingeben der Symbole der Zahl  $\rightarrow$  symbole[]

pot = 0

Wiederholen solange symbole[] länger als 0 bzw. []   ! leer

    entferne letztesSymbol  $\rightarrow$  sym

    sym<sup>pot</sup>  $\rightarrow$  potwert

    erg + potwert  $\rightarrow$  erg

    Erhöhe pot           !pot = pot + 1

Ausgeben erg



## 1.1.x.y.z. Rechnen im dualen Zahlensystem

### Addition

Summand 1	0	0	1	1
Summand 2	0	1	0	1
Summe(n-Wert)	0	1	1	10

### **Exkurs: Wie ging den noch die schriftliche Addition im Dezimalsystem?**

Im Zeitalter von Taschenrechnern und ewig verfügbaren Handy's bzw. Smartfon's gehen die – in der Grundschule gelernten mathematischen Methoden schnell den Weg ins Nirwana (bei uns Informatikern ins NIL).

Trotzdem wird es ab und zu doch mal wieder notwendig sein, einfach mal zwei Zahlen zu addieren.

$$1725 + 9562 = ?$$

An die Methodik / den Algorithmus werden sich die meisten noch erinnern. Zur Sicherheit berechnen wir mit den Taschenrechner od.ä. einmal vor, was raus kommen sollte.

$$1725 + 9562 = 11287$$

Wie ging das nun genau beim schriftlichen Addieren?

1. Notiere die Zahlen so untereinander, dass sie hinsichtlich des Stellensystems übereinstimmen (quasi: Komma über Komma). Ziehe einen Strich unter die Zahlen und notiere ein Plus-Zeichen vor die unterste Zahl.

$$\begin{array}{r} 1725, \\ + 9562, \\ \hline \end{array}$$

2. Beginne bei der niedrigsten Stelle

3. Addiere die Ziffern der gleichen Stellenposition bei allen Zahlen miteinander. Notiere die letzte Ziffer dieser Summe unter dem Strich ander zugehörigen Stellenposition.

$$\begin{array}{r} 1725, \\ + 9562, \\ \hline \end{array}$$

2. Beginne bei der niedrigsten Stelle

3. Addiere die Ziffern der gleichen Stellenposition bei allen Zahlen miteinander. Notiere die letzte Ziffer dieser Summe unter dem Strich ander zugehörigen Stellenposition.

$$\begin{array}{r} 1725, \\ + 9562, \\ \hline \end{array}$$

#### **Hilfe bei Vorzeichen:**

$$(+ ) + (+ ) = (+ )$$

$$(+ ) + (- ) = (+ ) - (+ ) = (+ ) \quad | \quad (- )$$

$$(- ) + (+ ) = (- ) \quad | \quad (+ )$$

$$(- ) + (- ) = (- )$$

begonnen wird, wie gewöhnlich ganz rechts

es wird die Werte-Tabelle genutzt (oder das logisch-mathematische Grundschul-Verständis) bei der Addition von 1 und 1 ergibt sich an der Position eine 0 und eine Position links ein Übertrag (weil die aktuelle Stelle ausgenutzt ist und die höherwertige Stelle benutzt werden muss) dieser wird in die Berechnung der nächsten Stelle mit einbezogen

		$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	dezimal
<b>Summand 1</b>			0	1	0	0	1	0	0	1	= 73
<b>Summand 2</b>			1	1	1	1	1	1	0	0	= 252
Übertrag			1	1	1	1					
<b>Ergebnis</b>		1	0	1	0	0	0	1	0	1	= 325

Bei Berechnungen in Computersystemen muss immer bedacht werden, dass diese endliche Größen für die zu verarbeitenden Zahlen haben (üblich: 8, 16, 32, 64, ... bit).

Selbst bei universellen Berechnern (einige Programmiersprachen (z.B. Python) oder CAS-Programme (z.B.: MuPAD)) begrenzt der Speicher (Hauptspeicher ev. mit Festplatte etc.) die Berechnungsmöglichkeiten. Rechnungen müssen / sollten immer auf Gültigkeit geprüft werden und passende Datentypen ausgewählt werden. Der gewählte Datentyp sollte immer reichlich Reserve nach oben und unten bieten!

## Subtraktion

Subtraktion nur begrenzt definiert, da es keine negativen dualen Zahlen gibt praktisch also eine Subtraktion, wie wir sie in der ersten Klasse kennen gelernt haben (als wir noch keine negativen Zahlen kannten)

Minuend	0	0	1	1
Subtrahend	0	1	0	1
Differenz(-Wert)	0	-1	1	0

		$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	dezimal
<b>Minuend</b>			1	1	1	1	0	1	0	1	= 245
<b>Subtrahend</b>			0	1	0	0	1	1	0	0	= 76
geborgt / Borrow-Bit						-1					
<b>Ergebnis</b>			1	0	1	0	1	0	0	1	= 169

## **Subtraktion über das Zweier-Komplement**

benutzt die interne Darstellung von negativen Zahlen in Computersystemen macht sich zu nutze, dass eine Subtraktion auch als Addition geschrieben werden kann

für das Beispiel:  $245 - 76$  lässt sich auch schreiben  $245 + (-76)$

negative Zahlen werden als Zweier-Komplement dargestellt

Der Minuend wird ev. durch eine führende Null erweitert. Diese steht quasi für das Plus-Zeichen.

Da nur eine kleinere Zahl (Subtrahend) von einer größeren (Minuend) abgezogen werden kann, muss die kleinere Zahl zuerst einmal auf die gleich Länge gebracht werden, d.h. auf der linken Seite werden die nicht-relevanten Nullen (0) notiert.

Wenn die kleinere Zahl keine führende Null hat, dann werden beide Zahlen um führende Nullen erweitert.

	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	dezimal
Minuend	0	1	1	1	1	0	1	0	1	= 245
Subtrahend			1	0	0	1	1	0	0	= 76
Subtrahend (erweitert)	0	0	1	0	0	1	1	0	0	= 76
Übertrag										
Ergebnis										

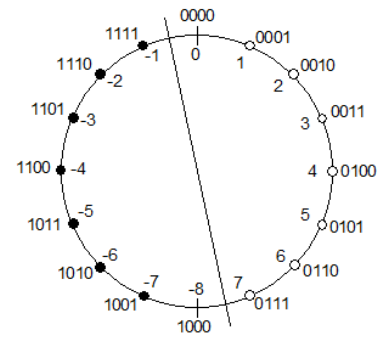
In der technischen Praxis muss dann natürlich die Verarbeitungsbreite der Rechen-Einheit bzw. des Systems (z.B.: 8, 16, 32 od. 64 bit) beachtet werden.

Die neue Zahl wird nun bit-weise negiert. Dadurch entsteht an der ganz linken Stelle eine **1**, die praktisch das Minus-Zeichen darstellt (Most-Significant-Bit, MSB). Damit erhält man erst einmal das Komplement.

Für das Zweier-Komplement muss nun noch eine 1 addiert werden.

Wie die technische Zahlen-Darstellung (zumindestens für ganze Zahlen) aussieht, kann der auf 4 bit vereinfachten Realisierung (rechte Abb.) entnehmen.

Wird also ein Nibble (eine 4-bit-Zahl) zur Darstellung von ganzen Zahlen genutzt, dann ergibt sich ein Wertebereich von -8 bis 7.



technischer Zahlenkreis  
Q: [www.info-wsf.de](http://www.info-wsf.de) (Ingo Höpping)

	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	dezimal
Minuend	0	1	1	1	1	0	1	0	1	= 245
Subtrahend			1	0	0	1	1	0	0	= 76
Zahlenerweiterung	0	0	1	0	0	1	1	0	0	= 76
Negation (Komplement)	1	1	0	1	1	0	0	1	1	
Addition von 1		0	0	0	0	0	0	0	1	
Übertrag							1	1		
Zweierkomplement	1	1	0	1	1	0	1	0	0	
Übertrag										
Ergebnis										

Im letzten Schritt wird nun die Addition von Minuend und Zweierkomplement des Subtrahenten berechnet:

	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	dezimal
Minuend	0	1	1	1	1	0	1	0	1	= 245
Subtrahend			1	0	0	1	1	0	0	= 76
Zweierkomplement	1	1	0	1	1	0	1	0	0	(-76)
Übertrag	1	1	1	1		1				
Ergebnis	<del>1</del>	0	1	0	1	0	1	0	0	= 169

Eine neue Ziffer (hier: **1**) auf der linken Seite wird ignoriert bzw. weggestrichen. Die restliche Zahl ist das Ergebnis. Ist das linke Bit eine 1, dann handelt es sich um eine negative Zahl, wenn diese dann im technischen System zugelassen ist.

In technischen System kann und kommt es bei Rechnungen mit ganzen Zahlen auch immer zu Bereichs-Überläufen. So wird z.B. bei der Addition (im 8-Bit-System) von 110 und 20 nicht etwa ein 130 berechnet, sondern das Ergebnis ist eine -126.

Programmierer müssen das beachten! Beim reinen Rechnen im Dualsystem hat das keine Bedeutung.

Wegen so einem Bereichs-Überlauf ist schon mal eine Ariane-5-Rakete explodiert, weil es wegen der Umstellung von Ariane 4 auf 5 zu größeren Zahlen kam. Der Computer berechnete auf einmal negative Werte (wegen des bereichs-Überlaufs) und das Regulations-System versuchte auszugleichen, was den Effekt nicht (bzw. nur unmerklich) reduzierte.

Bit-Sequenz	7-bit-Ganzzahl	
0111 1111	+127	
0111 1110	+126	
...		
0000 0010	2	
0000 0001	1	
0000 0000	0	
1111 1111	-1	
1111 1110	-2	
1111 1101	-3	
1111 1100	-4	
...		
1000 0100	-124	
1000 0011	-125	
1000 0010	-126	
1000 0001	-127	
1000 0000	-128	

**Aufgaben:**

- 1.
- 2.
- 3.

## Multiplikation

Faktor 1	0	0	1	1
Faktor 2	0	1	0	1
Produkt(-Wert)	0	0	0	1

Am Besten verfährt man bei der Multiplikation genauso, wie wir es vom schriftlichen Multiplizieren kennen. Die Zahlen werden über einen Strich als Aufgabe notiert. Der erste Faktor wird dann Stelle für Stelle mit den Bits des zweiten Faktors multipliziert und die Bit-Produkte jeweils Stellen-versetzt untereinander notiert. Das funktioniert praktisch sehr einfach, da die Multiplikation mit 1 immer die Sequenz des ersten Faktors ergibt. Ist der Bit-Faktor eine Null, dann kommt nur Null raus, was sich ebenfalls leicht notieren lässt. Am Ende müssen die versetzten Bit-Produkte "nur" noch addiert werden.

```

010101110 x 0110
-----
000000000
010101110
010101110
000000000
-----
10000010100
=====

```

### Aufgabe:

*Prüfen Sie die Multiplikation im dezimalen Zahlensystem!*

#### **Sonderfall: Multiplikation mit 2**

Da die doppelt so große Zahl (der Stellen-Wert) immer eine Position weiter links steht, lässt es sich leicht mit 2 multiplizieren, indem man die Ziffern eine Position nach links verschiebt und die frei werdende rechte Position ( $2^0$ ) mit **0** aufgefüllt.

	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	dezimal
<b>Faktor 1</b>		0	1	0	0	1	1	0	1	= 77
<b>1x Links-Verschiebung</b>	0	1	0	0	1	1	0	1	<b>0</b>	= 154

## Division

Dividend	0	0	1	1
Divisor	0	1	0	1
Quotient(en-Wert)	n.def.	0	n.def.	1

Auch die Division wird äquivalent zur schriftlichen Division durchgeführt. Über den Strich notiert man die Aufgabe. Nun wird der Divisor mit seiner führenden 1 unter die führende 1 des Dividenden positioniert. Man prüft nun, ob eine Subtraktion möglich ist. Wenn dies funktioniert, wird eine 1 auf der Ergebnisseite notiert und die Subtraktion ausgeführt. Geht die Subtraktion nicht, dann wird die nächste Stelle runter geholt und eine 0 auf die Ergebniss-Seite notiert. So wird weiter verfahren, bis alle Stellen abgearbeitet sind. Für noch offene Stellen des Dividenden (niederwertige Nullen) müssen je runtergeholter Null jeweils auch eine Null an die Ergebnis-Sequenz gehängt werden.

$$\begin{array}{r}
 11001000 : 1010 = 10100 \\
 \text{-----} \\
 -1010 \\
 \text{----} \\
 00101 \\
 \text{-----} \\
 1010 \\
 1010 \\
 \text{----} \\
 00 \\
 \text{-----} \\
 00
 \end{array}$$

Da Divisionen im Bereich der natürlichen bzw. ganzen sehr häufig mit gebrochenen Ergebnissen enden werden in den meisten Computersystemen die (ganzen) Zahlen vorher in Gleitkommazahlen umgewandelt und diese dann verarbeitet. Die Algorithmen für Gleitkommazahlen sind dann nicht mehr trivial.

### **Sonderfall: Division durch 2**

Divisor (Teiler) ist eine 2  
in Anlehnung an die Multiplikation mit 2 (Stellen-Verschiebung nach Links (Left-Shift, Links-Verschiebung)) wird die Division durch die Rechts-Verschiebung realisiert, die ganz rechte Stelle geht verloren (bzw. wird erste duale Dezimalstelle) und ganz links wird eine **0** aufgefüllt

	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$		dezimal
Dividend	1	0	1	1	0	1	1	0		= 182
1x Rechts-Verschiebung	<b>0</b>	1	0	1	1	0	1	1	<b>0</b>	= 91

### Aufgaben:

1.

#### für das gehobene Anspruchsniveau:

x. Geben Sie für die ersten vier dualen Nachkommastellen die Werte an!  
Stimmt das mit dem mathematischen Werten überein?

---

## 1.1.x.y.z. logische Operationen im dualen Zahlensystem

### NICHT-Operation (NOT, Negation)

auch Komplement  
Zeichen / Operator:  $\neg$

Operand	0	1
Ergebnis	1	0

### UND-Verknüpfung (AND, Konjunktion)

auch AND  
Zeichen / Operator:  $\wedge$  &  
bei Mengen auch  $\cap$

Operand 1	0	0	1	1
Operand 2	0	1	0	1
Ergebnis	0	0	0	1

### ODER-Verknüpfung (OR, Disjunktion)

auch OR  
Zeichen / Operator:  $\vee$   
bei Mengen auch  $\cup$

Operand 1	0	0	1	1
Operand 2	0	1	0	1
Ergebnis	0	1	1	1

### ENTWEDER-ODER-Verknüpfung (XOR, Antivalenz)

Antivalenz  
auch XOR, exklusives ODER  
Zeichen / Operator:  $\oplus$

Operand 1	0	0	1	1
Operand 2	0	1	0	1
Ergebnis	0	1	1	0

daneben von technischer Bedeutung: NAND (UND mit Eingangs-Negation, Inhibition) und NOR (ODER mit Eingangs-Negation, PEIRCE-Funktion)

praktisch kann mit 3 Basis-Funktionen alle 16 Funktionen realisieren

weitere logische Operationen: Kontradiktion, Identitäten, Äquivalenz (NXOR), Implikationen und Tautologie

## Übersicht über die logischen Operationen

	Operand 1	0	0	1	1			
Operation	Operand 2	0	1	0	1			
Kontradiktion (Falsch)		0	0	0	0			
Konjugation (AND)		0	0	0	1			
Inhibition (AND mit Eingangsnegation 2)		0	0	1	0			
Identität zum Operand 1		0	0	1	1			
Inhibition (AND mit Eingangsnegation 1)		0	1	0	0			
Identität zum Operand 2		0	1	0	1			
Antivalenz (XOR)		0	1	1	0			
Disjunktion		0	1	1	1			
PEIRCE-Funktion (NOR)		1	0	0	0			
Äquivalenz (NXOR)		1	0	0	1			
Negation von Operand 2		1	0	1	0			
Implikation aus Operand 2 (OR mit Eing.-Neg.)		1	0	1	1			
Negation von Operand 1		1	1	0	0			
Implikation aus Operand 1 (OR mit Eing.-Neg.)		1	1	0	1			
SHEFFER-Funktion (NAND)		1	1	1	0			
Tautologie (Wahr)		1	1	1	1			

### (Rechen-)Regeln bei logischen Operationen

abgeleitet von üblichen Gesetzen aus dem dezimalen Zahlensystem

#### **Assoziativitäts-Gesetz**

$$x \vee (y \vee z) = (x \vee y) \vee z$$

#### **Distributivitäts-Gesetz**

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

#### **Kommutativitäts-Gesetz**

$$x \vee y = y \vee x$$

#### **Vereinfachungs-Gesetz**

$$x \vee x = x$$



---

## Absorbtions-Gesetz

$$x \vee (x \wedge y) = x$$

## DEMORGANSche Gesetze

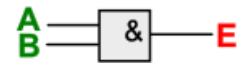
$$\neg x \vee \neg y = \neg(x \wedge y)$$

$$\neg x \oplus \neg y = x \oplus y$$

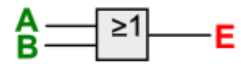
## Logik-Gatter – die technische Realisierung von Logik-Operationen

Symbole an DIN 40900 entlehnt (dort nicht grau ausgefüllt!)

UND-Gatter / AND-Gatter  
Konjugation



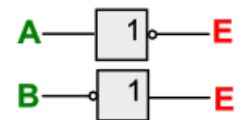
ODER-Gatter / OR-Gatter  
Disjunktion



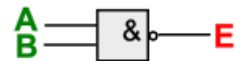
ExODER-Gatter / XOR-Gatter  
Antivalenz



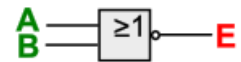
NICHT-Gatter / OR-Gatter  
Negation



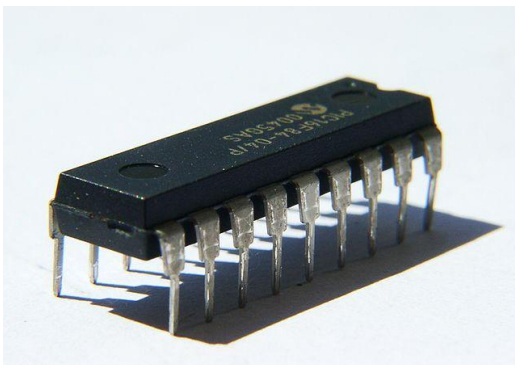
NICHT-UND-Gatter / NAND-Gatter  
SHEFFER-Funktion



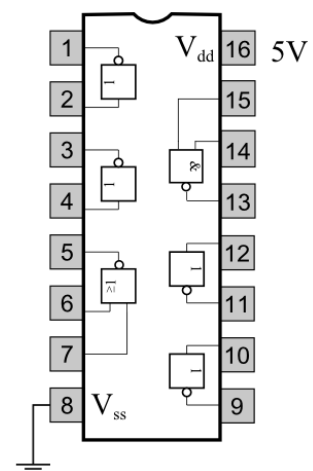
NICHT-ODER-Gatter / NOR-Gatter  
PEIRCE-Funktion



Logische Gatter werden zusammengefasst als Integrierte Schaltkreise (Integrated Circuit) produziert und in elektronischen Geräten verbaut. Eine der Krönungen dieser Bautechnik sind moderne Microprozessoren mit mehr als rund 40 Mrd. Transistoren in einem IC.

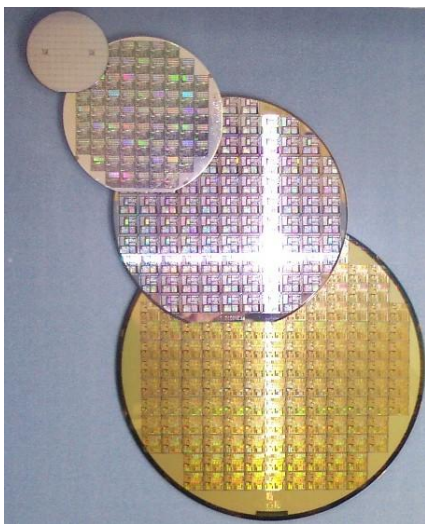


IC (Integrated Circuit) Integrierte Schaltung  
Q: de.wikipedia.org (Wollschaf)

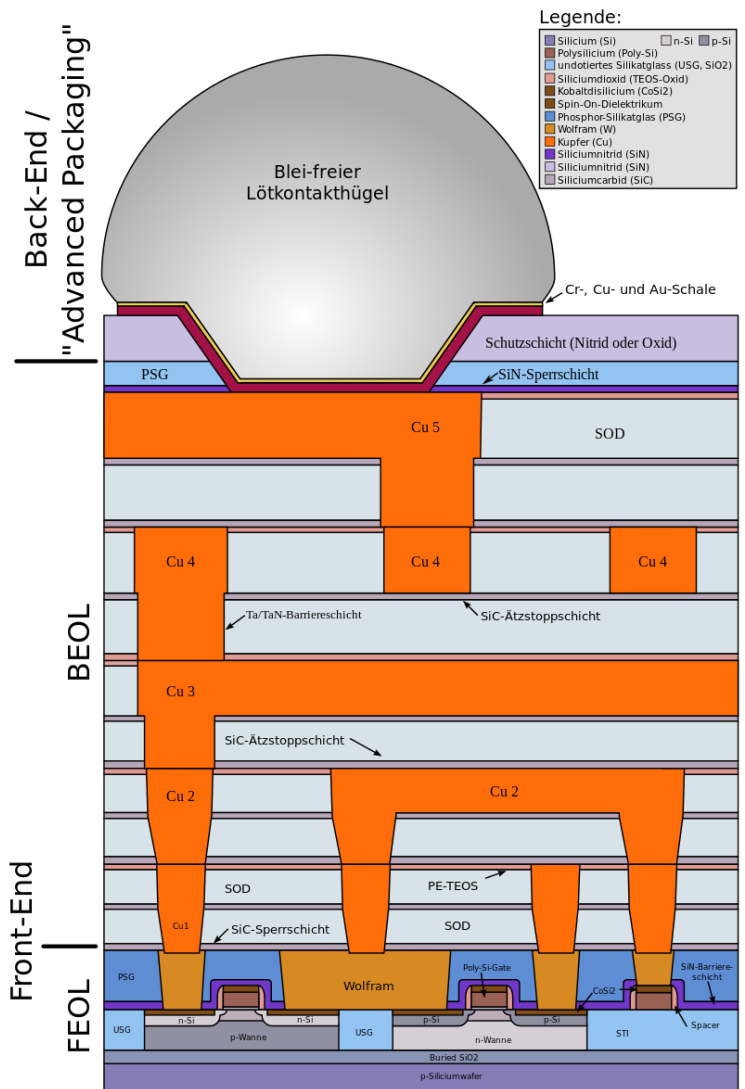


CD4572UB  
Integrierter CMOS-Baustein  
Texas Instruments CD4572UB  
Q: [https://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/ti\\_3\\_3\\_ger\\_web.html](https://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/ti_3_3_ger_web.html)

Jede einzelne Schicht wird über ein Bedampfungs-Verfahren auf die Silicium-Scheiben (Wafer) aufgetragen. Vorher wurde ein Photolack aufgetragen, dieser praktisch wie ein Dia belichtet und das Schalt-Muster so übertragen. Die nicht belichteten Stellen werden herausgeätzt. In die Lücken kann dann das spezielle Material aufgedampft werden. Ein Wafer enthält meist viele fertige Schaltungs-Platinen, die dann herausgebrochen werden und dann in ein Plastik- oder Keramik-Gehäuse verlötet werden.



Wafer unterschiedlicher Größe  
 Q: de.wikipedia.org (Stahlkocher + Saperaud-commonswiki)



Schichten-Struktur eines CMOS-Schaltkreises  
 Q: de.wikipedia.org ()

**Aufgaben:**

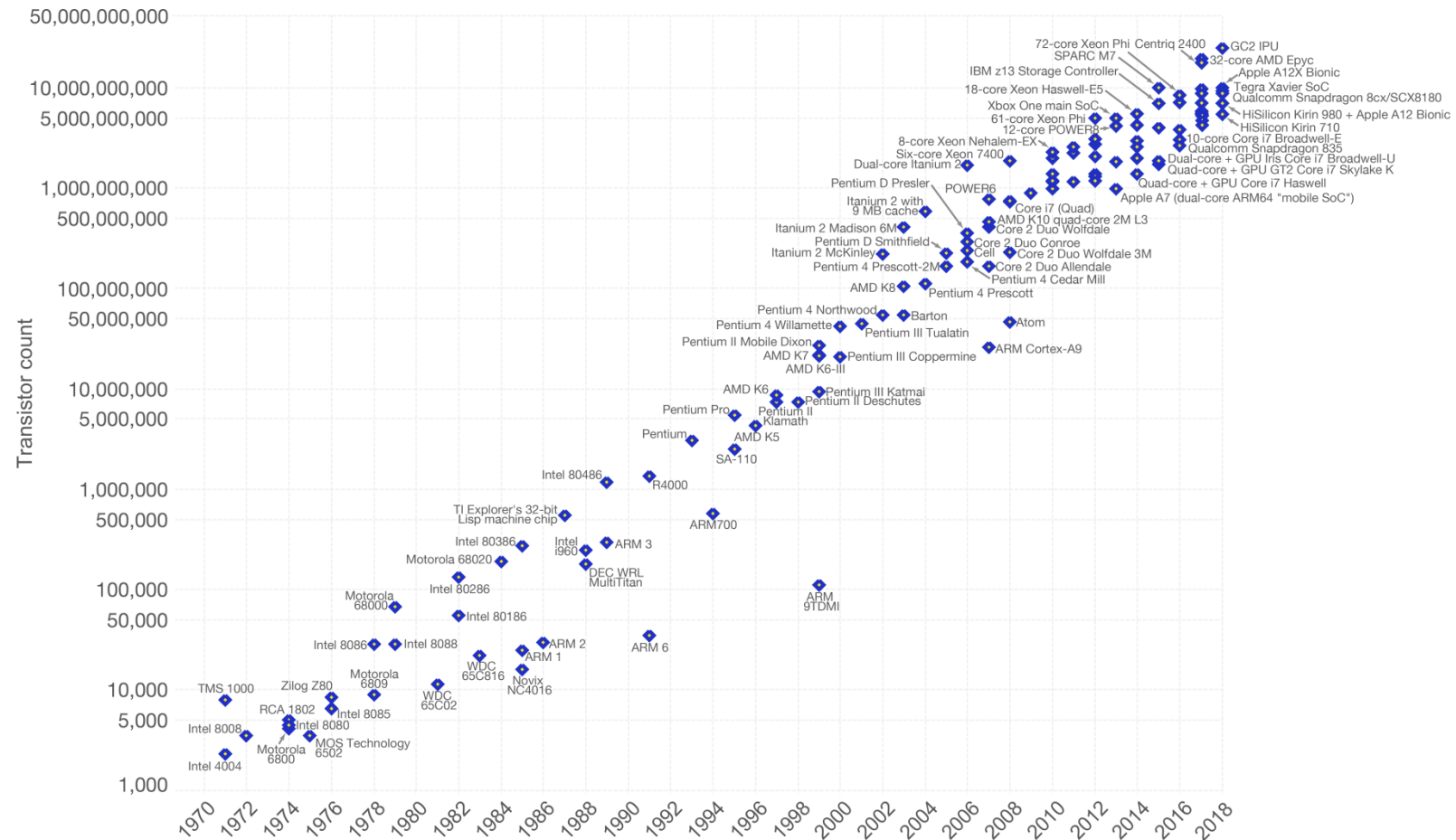
1. Auf der nächsten Seite ist ein Diagramm zum MOOREschen Gesetz abgebildet. Ist darauf wirklich ein exponentielles Wachstum zu sehen? Erläutern Sie Ihre Meinung!
- 2.

Das MOORESche Gesetz besagt, dass sich jedes Jahr die Anzahl der Transistoren auf einem IC verdoppeln würden. MOORE stellte diese Regel schon 1965 kurz nach der Erfindung der IC's auf. Die Verdopplungs-Zeit wurde dann kurze Zeit später von MOORE auf 2 Jahre korrigiert. Heute schwankt dieser Zeitraum zwischen 1 bis 2 Jahre.

In manchen Phasen kam es allerdings auch zu wesentlich langsameren Entwicklungen der Technik. Oft musste erst wieder eine neue Technologie entwickelt werden. So konnte man nicht mehr mit Licht bzw. UV-Licht belichten, sondern musste auf Laser wechseln, die extreme UV-Strahlung herstellen. Dafür war dann eine totale Umstellung aller Produktionsschritte und –Materialien (z.B. der Photolacke) notwendig.

## Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

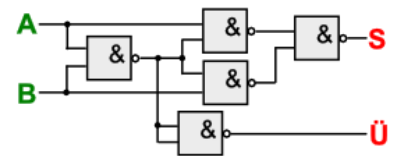
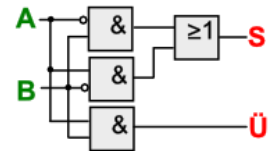
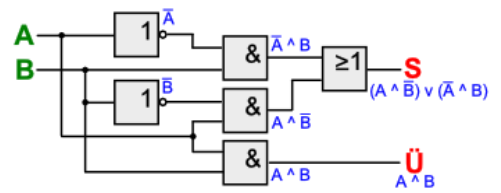
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia ([https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))  
 The data visualization is available at [OurWorldinData.org](https://www.ourworldindata.org). There you find more visualizations and research on this topic.

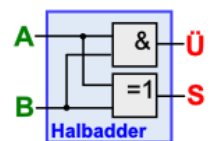
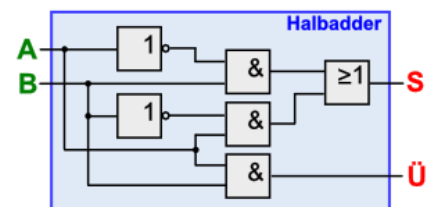
Licensed under CC-BY-SA by the author Max Roser.

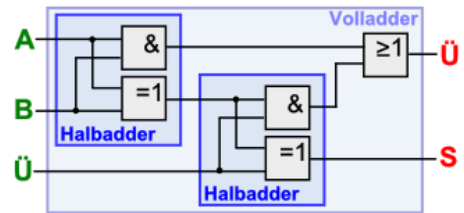
## Umsetzung in ein Rechenwerk



### Aufgaben:

1. Ergänzen Sie die logischen Ergebnisse nach den einzelnen Bausteinen!  
(siehe oben)
- 2.
- 3.





**Aufgaben für die gehobene Anspruchsebene:**

1. Überlegen Sie sich die Rechen-Funktionen (Addition, Substraktion und Multiplikation) für das Ternär-System, wobei die Zahlen 1, 0 und 1 verwendet werden sollen! (1 steht dabei für -1)
2. Überlegen Sie sich die grundlegenden Logik-Funktionen ("UND", "ODER" und "NICHT") für das Ternär-System, wobei die Zahlen 1, 0 und 1 verwendet werden sollen! (1 steht dabei für -1 und entspricht der Negation von 1)

LogicSim

Digital

## 1.1.x.y. das hexadezimale Zahlensystem

Nibble sind 4 bit auch Halbbyte genannt

Pos.	Pot.	dual (Nibble)	dez.
0	16 <sup>0</sup>	0001	1
1	16 <sup>1</sup>	0010	16
2	16 <sup>2</sup>	0100	4
3	16 <sup>3</sup>	0000 1000	8
4	16 <sup>4</sup>	0001 0000	16
5	16 <sup>5</sup>	0010 0000	32
6	16 <sup>6</sup>	0100 0000	64
7	16 <sup>7</sup>	0000 1000 0000	128
8	16 <sup>8</sup>	0001 0000 0000	256
9	16 <sup>9</sup>	0010 0000 0000	512
10	16 <sup>10</sup>	0100 0000 0000	1'024
11	2 <sup>11</sup>	0000 1000 0000 0000	2'048
12	2 <sup>12</sup>	0001 0000 0000 0000	4'092
13	2 <sup>13</sup>	0010 0000 0000 0000	8'192
14	2 <sup>14</sup>	0100 0000 0000 0000	16'384
15	2 <sup>15</sup>	0000 1000 0000 0000 0000	32'768
16	2 <sup>16</sup>	0001 0000 0000 0000 0000	65'536
17	2 <sup>17</sup>	0010 0000 0000 0000 0000	131'072
18	2 <sup>18</sup>	0100 0000 0000 0000 0000	262'144
19	2 <sup>19</sup>	0000 1000 0000 0000 0000 0000	524'288
20	2 <sup>20</sup>	0001 0000 0000 0000 0000 0000	1'048'576

dual / Nibble	hexadezimal	dezimal	octal
0000	0	0	00
0001	1	1	01
0010	2	2	02
0011	3	3	03
0100	4	4	04
0101	5	5	05
0110	6	6	06
0111	7	7	07
1000	8	8	10
1001	9	9	11
1010	A	10	12
1011	B	11	13
1100	C	12	14
1101	D	13	15
1110	E	14	16
1111	F	15	17

dual	hex.	dez.	dual	hex.	dez.	dual	hex.	dez.	dual	hex.	dez.
0000 0000	00	0	0100 0000	40	64	1000 0000	80	128	1100 0000	C0	192
0000 0001	01	1	0100 0001	41	65	1000 0001	81	129	1100 0001	C1	193
0000 0010	02	2	0100 0010	42	66	1000 0010	82	130	1100 0010	C2	194
0000 0011	03	3	0100 0011	43	67	1000 0011	83	131	1100 0011	C3	195
0000 0100	04	4	0100 0100	44	68	1000 0100	84	132	1100 0100	C4	196
0000 0101	05	5	0100 0101	45	69	1000 0101	85	133	1100 0101	C5	197
0000 0110	06	6	0100 0110	46	70	1000 0110	86	134	1100 0110	C6	198
0000 0111	07	7	0100 0111	47	71	1000 0111	87	135	1100 0111	C7	199
0000 1000	08	8	0100 1000	48	72	1000 1000	88	136	1100 1000	C8	200
0000 1001	09	9	0100 1001	49	73	1000 1001	89	137	1100 1001	C9	201
0000 1010	0A	10	0100 1010	4A	74	1000 1010	8A	138	1100 1010	CA	202
0000 1011	0B	11	0100 1011	4B	75	1000 1011	8B	139	1100 1011	CB	203
0000 1100	0C	12	0100 1100	4C	76	1000 1100	8C	140	1100 1100	CC	204
0000 1101	0D	13	0100 1101	4D	77	1000 1101	8D	141	1100 1101	CD	205
0000 1110	0E	14	0100 1110	4E	78	1000 1110	8E	142	1100 1110	CE	206
0000 1111	0F	15	0100 1111	4F	79	1000 1111	8F	143	1100 1111	CF	207
0001 0000	10	16	0101 0000	50	80	1001 0000	90	144	1101 0000	D0	208
0001 0001	11	17	0101 0001	51	81	1001 0001	91	145	1101 0001	D1	209
0001 0010	12	18	0101 0010	52	82	1001 0010	92	146	1101 0010	D2	210
0001 0011	13	19	0101 0011	53	83	1001 0011	93	147	1101 0011	D3	211
0001 0100	14	20	0101 0100	54	84	1001 0100	94	148	1101 0100	D4	212
0001 0101	15	21	0101 0101	55	85	1001 0101	95	149	1101 0101	D5	213
0001 0110	16	22	0101 0110	56	86	1001 0110	96	150	1101 0110	D6	214
0001 0111	17	23	0101 0111	57	87	1001 0111	97	151	1101 0111	D7	215
0001 1000	18	24	0101 1000	58	88	1001 1000	98	152	1101 1000	D8	216
0001 1001	19	25	0101 1001	59	89	1001 1001	99	153	1101 1001	D9	217
0001 1010	1A	26	0101 1010	5A	90	1001 1010	9A	154	1101 1010	DA	218
0001 1011	1B	27	0101 1011	5B	91	1001 1011	9B	155	1101 1011	DB	219
0001 1100	1C	28	0101 1100	5C	92	1001 1100	9C	156	1101 1100	DC	220
0001 1101	1D	29	0101 1101	5D	93	1001 1101	9D	157	1101 1101	DD	221
0001 1110	1E	30	0101 1110	5E	94	1001 1110	9E	158	1101 1110	DE	222
0001 1111	1F	31	0101 1111	5F	95	1001 1111	9F	159	1101 1111	DF	223
0010 0000	20	32	0110 0000	60	96	1010 0000	A0	160	1110 0000	E0	224
0010 0001	21	33	0110 0001	61	97	1010 0001	A1	161	1110 0001	E1	225
0010 0010	22	34	0110 0010	62	98	1010 0010	A2	162	1110 0010	E2	226
0010 0011	23	35	0110 0011	63	99	1010 0011	A3	163	1110 0011	E3	227
0010 0100	24	36	0110 0100	64	100	1010 0100	A4	164	1110 0100	E4	228
0010 0101	25	37	0110 0101	65	101	1010 0101	A5	165	1110 0101	E5	229
0010 0110	26	38	0110 0110	66	102	1010 0110	A6	166	1110 0110	E6	230
0010 0111	27	39	0110 0111	67	103	1010 0111	A7	167	1110 0111	E7	231
0010 1000	28	40	0110 1000	68	104	1010 1000	A8	168	1110 1000	E8	232
0010 1001	29	41	0110 1001	69	105	1010 1001	A9	169	1110 1001	E9	233
0010 1010	2A	42	0110 1010	6A	106	1010 1010	AA	170	1110 1010	EA	234
0010 1011	2B	43	0110 1011	6B	107	1010 1011	AB	171	1110 1011	EB	235
0010 1100	2C	44	0110 1100	6C	108	1010 1100	AC	172	1110 1100	EC	236
0010 1101	2D	45	0110 1101	6D	109	1010 1101	AD	173	1110 1101	ED	237
0010 1110	2E	46	0110 1110	6E	110	1010 1110	AE	174	1110 1110	EE	238
0010 1111	2F	47	0110 1111	6F	111	1010 1111	FA	175	1110 1111	EF	239
0011 0000	30	48	0111 0000	70	112	1011 0000	B0	176	1111 0000	F0	240
0011 0001	31	49	0111 0001	71	113	1011 0001	B1	177	1111 0001	F1	241
0011 0010	32	50	0111 0010	72	114	1011 0010	B2	178	1111 0010	F2	242
0011 0011	33	51	0111 0011	73	115	1011 0011	B3	179	1111 0011	F3	243
0011 0100	34	52	0111 0100	74	116	1011 0100	B4	180	1111 0100	F4	244
0011 0101	35	53	0111 0101	75	117	1011 0101	B5	181	1111 0101	F5	245
0011 0110	36	54	0111 0110	76	118	1011 0110	B6	182	1111 0110	F6	246
0011 0111	37	55	0111 0111	77	119	1011 0111	B7	183	1111 0111	F7	247
0011 1000	38	56	0111 1000	78	120	1011 1000	B8	184	1111 1000	F8	248
0011 1001	39	57	0111 1001	79	121	1011 1001	B9	185	1111 1001	F9	249
0011 1010	3A	58	0111 1010	7A	122	1011 1010	BA	186	1111 1010	FA	250
0011 1011	3B	59	0111 1011	7B	123	1011 1011	BB	187	1111 1011	FB	251
0011 1100	3C	60	0111 1100	7C	124	1011 1100	BC	188	1111 1100	FC	252
0011 1101	3D	61	0111 1101	7D	125	1011 1101	BD	189	1111 1101	FD	253
0011 1110	3E	62	0111 1110	7E	126	1011 1110	BE	190	1111 1110	FE	254
0011 1111	3F	63	0111 1111	7F	127	1011 1111	BF	191	1111 1111	FF	255
0100 0000	40	64	1000 0000	80	128	1100 0000	C0	192	1 0000 0000	100	256



---

## Links

<https://www.translatorscafe.com/unit-converter/DE/numbers/3-23/decimale-base%2020/>

(Umrechnungen der Zahlensysteme)

Grenzen von Zahlen-System  
z.B. Zweier-Potenzen in Excel

Beispiel in Python  $0.2 + 0.1$  ergibt  $0.300000000004$   
→ Fließkomma-Zahlen-Darstellung in Rechner-Systemen

führend	folgend (Einer, 16 <sup>u</sup> )																		
HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	00	000	0000
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	0	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	256	4096	65536
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	512	8192	131072
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	768	12288	196608
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	1024	16384	262144
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	1280	20480	327680
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	1536	24576	393216
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	1792	28672	458752
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	2048	32768	524288
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	2304	36864	589824
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	2560	40960	655360
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	2816	45056	720896
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	3072	49152	786432
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	3328	53248	851968
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	3584	57344	917504
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	3840	61440	983040
10	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	4096	65536	1048576

## 1.2. Aufbau von Datenverarbeitungsanlagen

Datenverarbeitungsanlage = Daten-verarbeitendes System

### **Definition(en): Datenverarbeitungsanlage**

Eine Datenverarbeitungsanlage (EDVA) ist ein System, das mittels Vorschriften (Befehlen, Anweisungen) mit Informationen (Daten) umgeht.

Ein Rechner / Computer / eine EDVA ist ein technisches Gerät(e-System) zum Abarbeiten von Programmen mit denen Daten zielgerichtet manipuliert werden sollen.

### **Definition(en): Hardware**

Hardware ist die technische (mechanische und elektr(on)ische) Ausstattung einer Datenverarbeitungsanlage (DVA).

Hardware sind die materiellen Bestandteile eine Datenverarbeitungsanlage.

### **Definition(en): Software**

Software sind die in der Datenverarbeitungsanlage benutzten Programme und Daten.

Software sind die immateriellen Bestandteile einer Datenverarbeitungsanlage.

Software sind die nicht-technischen Bestandteile einer Datenverarbeitungsanlage.

Software basiert immer auf Hardware, innerhalb derer die Information durch verschiedene Systemzustände und / oder Strukturen repräsentiert wird

### **FLYNNsche Klassifikation**

klassische Rechner (PCs usw.) besitzen SISD-Architekturen  
Single Instruction, Single Data

	<b>Single Instruction</b>	<b>Multiple Instruction</b>
<b>Single Data</b>	<b>SISD</b> (PCs, ... (VON-NEUMANN-Rechner; Harvard-Rechner))	<b>MISD</b> (wenige Großrechner bzw. Supercomputer (z.B. Schachcomputer))
<b>Multiple Data</b>	<b>SIMD</b> (Graphik- und Video-Verarbeitung, Großrechner, Supercomputer)	<b>MIMD</b> (Großrechner, Supercomputer, verteiltes Rechnen)

<b>Definition(en): Informatik-System / informatisches System</b>
Ein Informatik-System ist eine Zusammenstellung von Hardware-, Software- und Netzwerk-Komponenten zur Lösung eines Anwendungs-Problems.

Parallel-Verarbeitung  
mehrere unabhängig voneinander arbeitende Rechensystem (praktisch VON-NEUMANN-Rechner, die über ein Netzwerk oder Bus-System miteinander kommunizieren  
besonders häufig im Graphik-Bereich benutzt, weil dort z.B. für Millionen von Pixeln die gleichen berechnungen durchgeführt werden müssen und das unabhängig vom Nachbar-Pixel  
Verwaltungs- und Steuer-Anteil steigt überproportional mit der Anzahl der parallelen Einheiten

<b>Definition(en): Programm</b>
Ein Programm ist eine Folge von (auf die Hardware und ihre Arbeitsweise zugeschnittene) Arbeitsvorschriften zum Lösen einer Aufgabe.
Ein Programm ist die Hardware-spezifische Umsetzung eines Algorithmus.

<b>Definition(en): Information</b>
Informationen sind Daten die ein informatisches System zur Erfüllung seiner Aufgabe braucht.

<b>Definition(en): Datum</b>
Ein Datum (Mehrzahl: Daten) ist die Information, die mittels eines Programms benutzt wird.

---

**Biographie: Konrad ZUSE (1910 - 1995)**

--

## 1.2.1. Grundelement Speicher

Speicher besteht aus Speicherzellen, die praktisch hintereinander / übereinander angeordnet sind

typische Speicherzelle sind 1 Byte breit – besitzen also 8 bit

in modernen PC's sind die Speicher breiter organisiert – meist 16 oder 32 bit  
es werden mehrere Speicherzellen gemeinsam gelesen / geschrieben  
Byte, Word (2 Byte), DoubleWord (2 Word = 4 Byte), Quadword (2 DWord = 8 Byte)

jede Speicherzelle hat eine einmalige Adresse

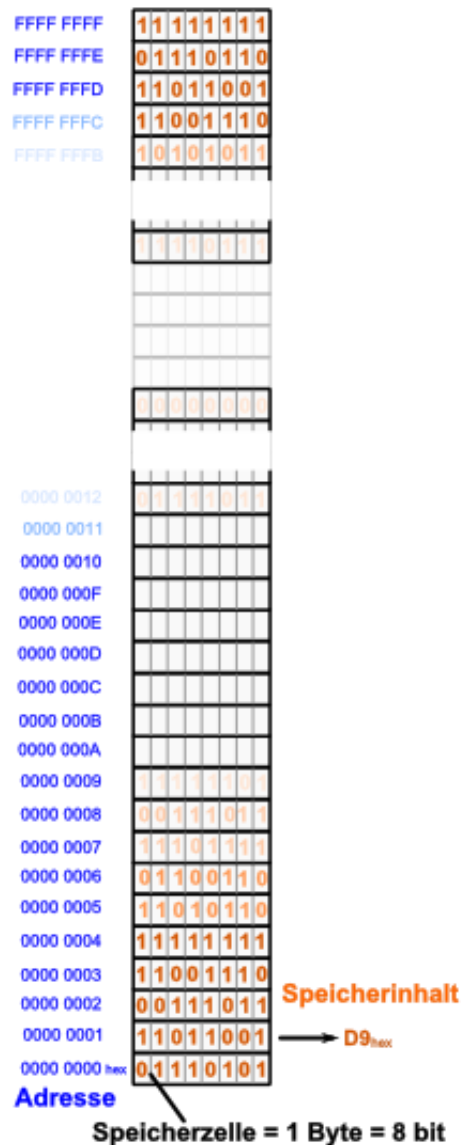
beginnend bei 0H fortlaufend bis zum Ende des adressierbaren Bereichs, der sich aus den Digitalstellen des Adress-Busses ergibt

8 bit Adress-Bus lässt nur 256 Byte zu  
16 bit ermöglicht 64 KiB (Kibi Byte) = 65'535 Byte

32 bit kann 4'294'967'295 Byte (4 GiB)  
moderne 64 bit-Systeme könnten mit 18'446'744'065'119'617'025 Byte = 16'000'000 TiB = 16'000 PiB = 16 EiB richtig viel abspeichern

Betriebssystem muss die Adress-Breite auch verarbeiten können

da liegt derzeit der begrenzende Faktor neben physikalischen Grenzen derzeit die einprogrammierten Grenzen entscheidend



Speicher muss nicht vollständig sein

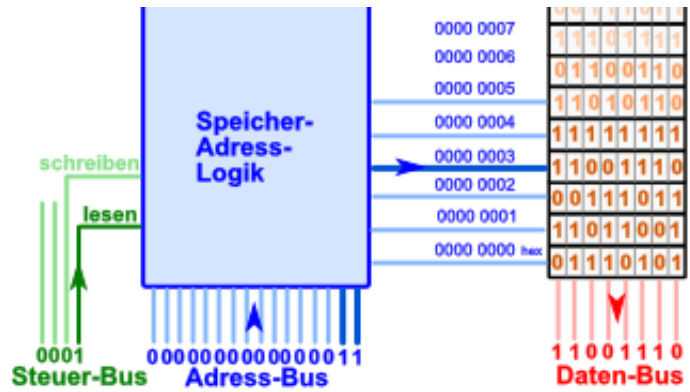
z.B. BIOS-Speicher liegt am Ende des physikalisch erreichbaren / adressierbaren Speicher-Bereichs

extra reservierte Speicher-Bereiche z.B. für Graphik-Karte, Tastatur-Zwischenspeicher (Tastatur-Cache), ...  
vom Computer-Bautyp abhängig

über die Leitungen des Adress-Bus wird die gültige Adresse eingestellt

liegt auf der Lese-Leistung (Steuer-Bus) ein Signal, dann wird adressierte Speicherzelle zum Kopieren der Daten auf den Daten-Bus angeregt

bei Schreib-Befehl wird der Inhalt des Daten-Bus in die adressierte Speicherzelle geschrieben



### Aufgaben:

1. Was bedeuten die Speichergrößen MiB, GiB und TiB?
- 2.

- KibiByte
- MebiByte
- GibiByte
- TebiByte
- GibiByte
- ExbiByte
- PebiByte
- ExbiByte
- ZebiByte
- YobiByte
- RobiByte
- QuebiByte

## 1.2.2. VON-NEUMANN-Architektur

1945

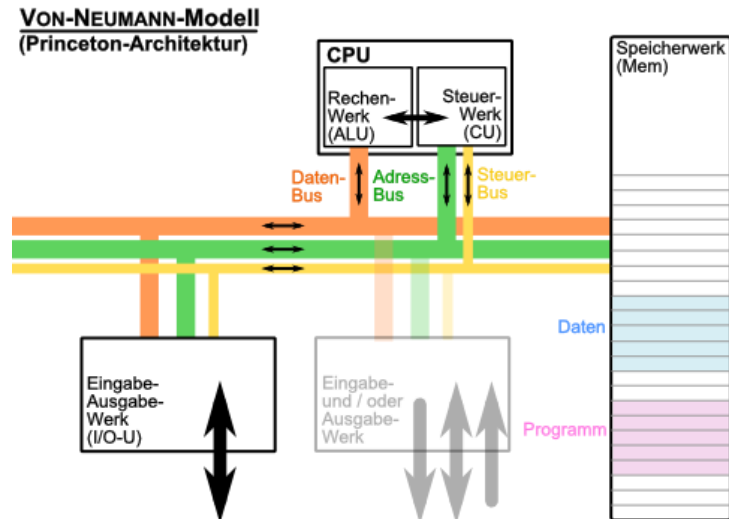
damals gab es quasi noch gar keine Rechner im heutigen Sinn

Rechner waren fest mit Programm versehen, entweder durch Hardware-Verschaltung oder über Lochkarten-Befehle und –Daten

John VON NEUMANN war Mathematiker aus Österreich-Ungarn, später dann in die USA emigriert

gleichzusetzen mit Princeton-Architektur (an der Princeton University entwickelt)

basiert auf Idee von Konrad ZUSE



### Prinzipien des VON-NEUMANN-Modells

- Gesamtsystem besteht aus 5 Funktionseinheiten (Werken)**  
 Rechenwerk, Steuerwerk, Speicherwerk, Eingabewerk, Ausgabewerk
- alle Werke sind über einen (zentralen) Bus miteinander verbunden**  
 parallele Adress-, Daten und Steuer-Leitungen (= Bus)
- die Zentraleinheit (CPU) arbeitet Taktgesteuert**  
 kHz bis GHz
- die interne Signalmenge ist binär kodiert**  
 0 od. 1
- im Rechner werden Worte fester Länge verarbeitet**  
 z.B. 4, 8, 16, 32, ... bit
- der Hauptspeicher besteht aus fortlaufend adressierten (Speicher-)Worten**  
 Speicherinhalt nur über Speicheradresse zugänglich
- System ist frei programmierbar**  
 Programme werden extern eingegeben und intern verarbeitet
- System ist hinsichtlich der verarbeiteten Daten universell**
- Zentraleinheit verarbeitet Befehls- und Datenworte sequenziell (hintereinander)**



---

in der theoretischen Informatik wird eine VON-NEUMANN-Rechner zu den TURING-Maschinen oder –Automaten gezählt (→ Automaten-Theorie → endliche Automaten)

im gemeinsamen Speicher befinden sich sowohl Daten als auch ein Programm

die Begriffe Rechner, Architektur und Modell werden hier äquivalent verwendet

Gegenstück ist Harvard-Architektur ()

### **Definition(en): VON-NEUMANN-Architektur**

Die VON-NEUMANN-Architektur ist ein Referenzmodell der Datenverarbeitung, bei der ein von mehreren Bestandteilen (Eingabe, Verarbeitung, Ausgabe) gemeinsam genutzter Speicher sowohl die Programme als auch die Daten enthält.

Ein VON-NEUMANN-Rechner ist ein Schaltungs- / Geräte-Konzept bei dem die vier Grundgeräte (Werke) (Rechenwerk, Steuerwerk, Speicherwerk und Ein-Ausgabe-Werk) über einen gemeinsam benutztes Bus-System (Adress-Bus, Daten-Bus, Steuer-Bus) verbunden sind und miteinander kommunizieren.

Speicher ist 1-dimensional strukturiert, praktisch ein Stapel aus Speicherzellen, die von Adresse 0 bis x (meist System- oder Hardware-Grenze) durchnummeriert sind

### **System-Ablauf:**

1. Programm (Befehle) befinden sich im Speicher
2. der Befehlzähler zeigt auf nächsten Befehl
3. Befehl (auf den (/ dessen Adresse) der Befehlszeiger zeigt) wird gelesen
  - a) ev. werden Daten eingelesen (Speicher → Register)
  - b) Befehl wird ausgeführt (Verknüpfung von Daten, Fällen von Entscheidungen, Berechnungen von Sprüngen), ev. werden Daten (z.B. Befehlsergebnisse) zurückgespeichert (Register → Speicher)
  - c) Befehlszähler verändert (normal um eins erhöht; bei Sprungbefehlen auch in größeren Schritten)
4. weiter bei 3.

ein minimales Programm muss im System fest eingespeichert sein (ROM, EEPROM, ...) (belegt meist bestimmte Speicherzellen (typisch ab Adresse 0)) ev. nur Sprung zu eigentlichem Arbeitssystem: z.B. BIOS (Basic Input/Output System)  
BIOS auf PCs liegt am oberen Ende des Speicher-Bereiches  
wenn Nutzer vom BIOS sprechen dann meinen sie das Konfigurations-Programm für das BIOS, das seine Konfiguration im cMOS-Speicher ablegt (Batterie-gestützt)  
damit lädt das System dann kaskadenartig übergeordnete Programme (z.B. Firmware (z.B. Treiber od. Hardware-interne (ROM-)Speicher), Betriebssystem, Anwenderprogramm)

auf modernen Rechnern durch UEFI (Unified Extensible Firmware Interface) ersetzt / erweitert

ist erweiterbare universelle und standardisierte Schnittstelle zwischen Firmware und Betriebssystem

---

in modernen Rechnern ist das VON-NEUMANN-Modell immer noch der zentrale Kern, aus der Sicht der Nutzer-Software bleibt dies auch so  
aus Hardware-Sicht ist VN-Modell teilweise aufgehoben oder hierarchisiert  
dazu kommen Tendenzen zur Parallelisierung und Virtualisierung, die auch auf Software-Seite das allgemeine Modell bröckeln lassen

VON-NEUMANN-Flaschenhals (von Neumann bottleneck)

ist das Bus-System, wenn eine Komponente / ein Werk bzw. eine Kombination von Quellwerk und Zielwerk darauf zugreift, dann ist er quasi für die anderen gespeert  
es kann immer nur eine Sache zur Zeit erledigt werden (die Einfachheit wird zum Problem)  
die Einfachheit wurde aber ursprünglich der Komplexität vorgezogen, weil sie damals nicht beherrschbar war

das VN-Modell hat deshalb so lange hervorragend funktioniert, weil die CPUs die komplexesten, Leistungsschwächsten und teuersten Bestandteile eines Computer waren  
heute gleichen sich die Komplexitäten, Leistungen und Preise stärker an (an den Grenzen des derzeit machbaren)

derzeit (2015) ist eher der Speicher das begrenzende Element (meist zu langsam); schnellerer Speicher zu teuer und deshalb zur als relativ kleiner Cache (Zwischenspeicher) benutzt

## **Grobaufbau einer CPU**

Breite der verschiedenen Speicher, Register und Zähler von der CPU-Architektur-Breite bestimmt

die meisten Speicher und Zähler haben die Breite der Architektur, dazu kommen einige mit doppelt so großer Breite

man unterscheidet Byte, Word und

Register als Speicher für Argumente, Ergebnisse und Zwischenergebnisse  
quasi Variablen-Speicher

RISC und CISC

Arithmetik-Logik-Einheit verarbeitet Microcode, der sich aus den Befehlen ableitet

man unterscheidet 1-Byte-, 2- und 3-Byte- Befehle

die 1-Byte-Befehle erfüllen bestimmte Operationen z.B. mit vorbelegten Zählern oder Registern

bei 2-Byte-Befehlen ist das 2. Byte häufig mit einem Argument belegt, es gibt aber auch komplexe Befehle, die erst mit dem 2. Byte vollständig sind

dies gilt z.B. bei Block-Befehlen so, die ganze Adress- oder Register-Blöcke mit einem Mal bearbeiten können

bei 3-Byte-Befehlen sind häufig 2 Argumente im 2. und 3. Byte folgend

VON-NEUMANN-Befehls-Zyklus  
Arbeits-Schrittfolge beim Erledigen eines Prozessor-Befehls

Schritt	Bezeichnung	Inhalt / Arbeits-Leistung
1	<b>(Instruction-) FETCH (IF)</b>	Befehls-Aufruf Befehl wird aus der Speicher-Adresse geladen, die in Befehls-Register (Befehls-Zähler) steht
2	<b>(Instruction-) DECODE (ID)</b>	Dekodierung Befehlszähler um die Befehls-Länge erhöht das Steuerwerk schaltet die zum Befehl gehörenden Logik-Schaltungen ein (bzw. gibt Daten-Leitung dahin frei)
3	<b>FETCH OPERANDS (FO)</b>	Operanden-Aufruf aus dem Speicher oder anderen Registern werden zusätzliche Operanden (Daten) geladen (auf die Logik-Schaltungen geleitet)
4	<b>EXECUTE (Execution Stage) (EX)</b>	Befehls-Ausführung Rechen-Logik arbeitet; Befehlszähler wird erhöht bzw. bei Schleifen / Sprüngen auf anderen Wert (neue Befehls-Adresse) gesetzt
5	<b>WRITE BACK (WB)</b>	Zurückschreiben (des Ergebnisses) ev. werden die Ergebnisse aus der Logik-Schaltung in Register oder Speicherzellen geschrieben

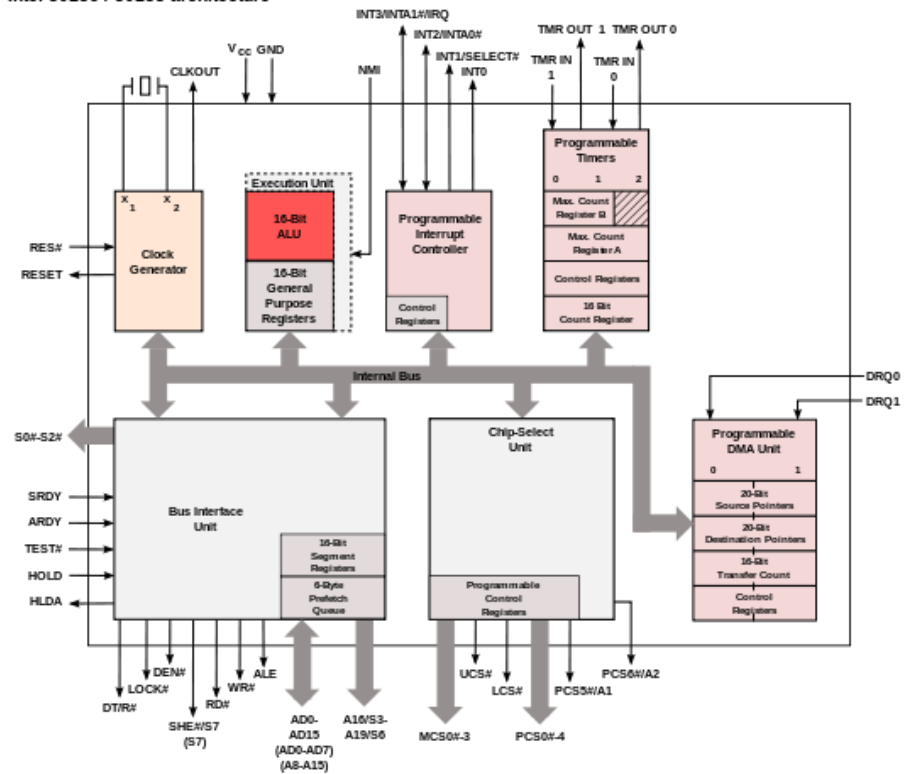
je nach Befehls-Typ und Daten-Menge werden für einen Befehl minimal ein Takt und maximal 5 Takte gebraucht  
die 5-stufige Prozessor-Taktung ist eher allgemein gemeint und theoretisch.  
In der Praxis sind die Prozessoren eher 4-stufig aufgebaut. Dabei fallen die Takte 2 und 3 meist zu einem zusammengefasst bzw. Arbeits-teilig ausgeführt  
in der ersten Takt-Hälfte erfolgt die Dekodierung und in der zweiten das Nachladen von Operanden

die meisten aktuellen Prozessoren arbeiten in einer Mischform zwischen VON-NEUMANN und Harvard-Architektur  
Daten und Befehle werden intern getrennt behandelt; nach außen hin (extern) aber in einem (gemeinsamen) Speicher abgelegt  
die Schritte von zwei Befehlen laufen nicht direkt hintereinander ab, sondern werden nur leicht Takt-versetzt abgearbeitet → Pipelining  
weiterhin erfolgt parallele Abarbeitung von Befehlen → Ports  
(bei modernen Prozessoren können so z.B. 40 Befehle gleichzeitig in der EXECUTE-Phase sein)

der Urvater (ab 1978) der heute weit verbreiteten Pentium®-Prozessoren usw. war der legendäre intel-8086  
mit 29'000 Transistoren, maximal 5 MHz, einer Verarbeitungsbreite von 16 bit und 14 Registern ist mit den heutigen Prozessor-Boliden nicht mehr zu vergleichen  
ungefähr 100 Befehle verfügbar

in den nächsten Jahren folgten dann immer Leistungs-stärkere, schnellere und universellere Prozessoren  
in der Nachfolge des 8086 waren dass z.B.:

Intel 80186 / 80188 architecture



Architektur des intel 80186  
Q: de.wikipedia.org (Appaloosa)

**Aufgaben:**

1. Ein Schüler hat in einer Klausur behauptet, ein Prozessor – wie z.B. der intel 80186 – wäre ein VON-NEUMANN-Rechner. Setzen Sie sich mit dieser Behauptung auseinander!
2. Stellen Sie von den Mikroprozessoren der intel-x86-Reihe die maximale Prozessorgeschwindigkeit und den (physikalisch) adressierbaren Speicher gegen das Ersterscheinungsjahr graphisch dar! Leiten Sie Tendenzen / Regeln ab!
3. Recherchieren Sie die ungefähren Transistoren-Zahlen und die Ersterscheinungsjahre der verschiedenen Mikroprozessoren der intel x86-Reihe und stellen Sie die graphisch gegen die Jahreszahl dar! Leiten Sie eine oder mehrere Tendenzen ab!

intel-Name	ab Jahr	Datenbus bit	Adressbus bit	max. Speicher (adressierbar)	Registerbreite bit	bis MHz	x	L1-Cache, L2-Cache, L3-Cache; max.	Kerne max.	
8086	1978	8	16		8 / 16	10		-	1	
80186	1982	16	20	1 MiB	8 / 16 / 32	16			1	
80286	1982	16	24	16 MiB	8 / 16 / 32	20			1	
80386	1985	32 / 16 (SX)	32 / 24 (SX)	4 GiB / 16 MiB (SX)	8 / 16 / 32 / 64	33			1	
80486	1989	32	32	4 GiB	8 / 16 / 32 / 64	50			1	
Pentium® (I) P5 (80586)	1993	64	32	4 GiB	8 / 16 / 32 / 64	66 / 200			1	RISC
Pentium Pro® (P6)	1995	64	36	64 GiB		200		8 + 8 KiB 1'024 KiB	1	
Pentium II®	1997	64	32	4 GiB		450		2'048 KiB	1	
Celeron	1998	64	32	4 GiB		2'500			1	
Pentium III®	1999	64	32	4 GiB		1'400			1	
Pentium 4®	2000	64	32	4 GiB					1	
Pentium M	2002	64	32	4 GiB		2'260			1	
Core 2 Duo	2006					3'300		6 MiB	2	
Core 2 Quad	2007	64	36	64 GiB		3'000			2x 2	
Core i7	2008	3x 64	36	64 GiB		3'330		32 KiB 256 KiB 8 MiB	4	
intel Atom®	2008	64	36	144 GiB		2'130		32 KiB 512 – 1'024 KiB	2 - 4	
i5	2009	64				3'500			4	
i7	2009	64				3'900			4 - 12	
i3	2010	64				3'400			2	

sehr vereinfachte Rechenformeln

$$\frac{\text{Anzahl\_Befehle}}{1\text{ s}} = \text{TaktFrequenz [Hz]}$$

funktioniert nur bei Ein-Takt-Befehlen  
(z.B. NOP)

$$\frac{\text{Anzahl\_Befehle}}{\text{TaktFrequenz [Hz]}} = \text{Zeit [s]}$$

vereinfacht für CPU- / Maschinen-Befehle mit den 5 klassischen Takten

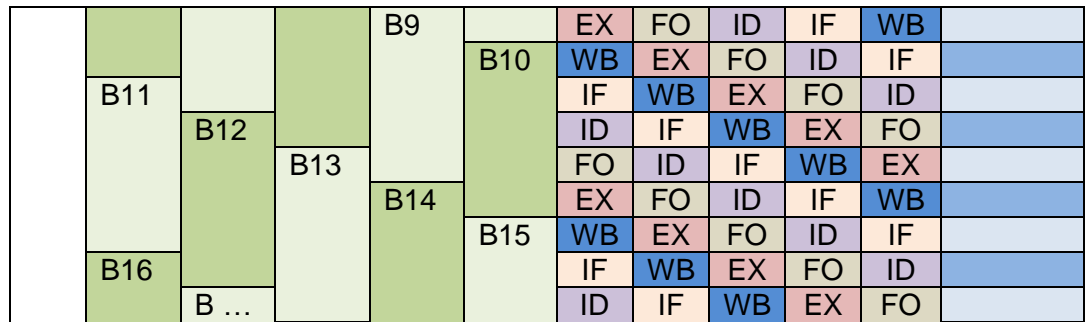
		Befehls-Takt		System-Takt		
Programm ...	Befehl 1		instruction fetch	1 Takt	5 Takte	
			instruction decode	1 Takt		
			fetch operand	...		
			execute			
			write back			
	Befehl 2		instruction fetch			5 Takte
			instruction decode			
			fetch operand			
			execute			
			write back			
	Befehl 3		instruction fetch			5 Takte
			instruction decode			
			fetch operand			
			execute			
			write back			
	Befehl ...		instruction fetch			...
			instruction decode			

### Pipelining

Für jeden Abarbeitungs-Teil eines Maschinen-Befehls sind andere Teile der CPU zuständig. Während der Arbeit eines Befehls-Auswerter sind also immer mehrere CPU-Teile unbenutzt. Es ist also sinnvoll, diese Teile immer schon mit nachfolgenden Befehlen zu beschäftigen. Die leicht verschobene – quasi fast parallele – Abarbeitung mehrerer Prozessor-Befehle nebeneinander nennt man Pipelining.

idealisiert für unabhängige Folge von Maschinen-Befehlen mit 5 Takten

		Befehls-Pipeline's					Prozessor-Pipelines's					System-Takt
Programm ...	B1					IF						1 Takt
		B2				ID	IF					1 Takt
			B3			FO	ID	IF				...
				B4		EX	FO	ID	IF			
					B5	WB	EX	FO	ID	IF		
	B6				IF	WB	EX	FO	ID			
		B7			ID	IF	WB	EX	FO			
			B8			FO	ID	IF	WB	EX		



Praktisch gibt es aber Abwandlungen und Probleme.

Befehle haben unterschiedliche Längen (Anzahl von Takten für ihre Abarbeitung)

eigentlich werden weniger Pipeline's im Prozessor gebrauch

bei Sprung-Befehlen (z.B. nach Vergleichen) ist eine Vorausschau (statistisch) zu 50 % umsonst; hier kann durch geeignete Compiler eine Optimierung erfolgen (normaler Schleifen-Körper in de Pipeline, Abbruch-Szenario dann mit selteneren unbenutzter Vorausschau, die verworfen wird)

ein Befehl in einer anderen Pipeline braucht ein Ergebnis aus einer vorlaufenden Pipeline, die aber noch keine Daten zurückgeschrieben hat; hier bauen moderne Compiler NOP-Befehle ein

hier ist Pipelining dann eher nachteilig (höhere Prozessor-Preise für Prozessoren, deren Leistungs-Potentiale nicht ausgenutzt werden können)

### Exkurs: Prozessoren von morgen?

"Rock Creek" (2009) ist ein sogenannter Single-Chip-Cloud-Prozessor, der aus 4 Gruppen a' 6 Doppelkern-Prozessoren-Einheiten (also 48 Kerne) zusammengesetzt ist. Intern neben Bus-Systemen wird hauptsächlich über Messages kommuniziert. Dafür sind auf dem Chip auch noch 24 Router integriert. Insgesamt sind 1,3 Milliarden Transistoren verbaut.

Praktisch kann jeder Kern mit einem eigenen Betriebssystem booten. Derzeit wird ein angepasstes Linux verwendet.

beobachtet man die Aktivitäten der CPU genauer, dann können die folgenden feiner gegliederten Abfolgen (Micro-Anweisungen) beobachtet werden

Daten (Anweisung) in Befehlsspeicher einlesen	<ol style="list-style-type: none"> <li>1. Adresse auf dem Adressbus einstellen</li> <li>2. Eingabeleitung auf Steuerbus aktivieren</li> <li>3. Daten vom Datenbus (in Befehlsspeicher) übernehmen</li> </ol>
Anweisung ausführen	<ol style="list-style-type: none"> <li>1. Befehlslogik einstellen (Anweisung decodieren)</li> <li>2. Befehlslogik aktivieren</li> <li>3. Befehls(adress)zähler um 1 erhöhen</li> </ol>
Daten eingeben	<ol style="list-style-type: none"> <li>1. Eingabeadresse auf dem Adressbus einstellen</li> <li>2. Eingabeleitung auf Steuerbus aktivieren</li> <li>3. Daten vom Datenbus übernehmen</li> </ol>
Daten in Speicher schreiben	<ol style="list-style-type: none"> <li>1. Adresse auf dem Adressbus einstellen</li> <li>2. Daten auf den Datenbus legen</li> <li>3. Schreibleitung auf Steuerbus aktivieren</li> </ol>
Daten in Register einlesen	<ol style="list-style-type: none"> <li>1. Adresse auf dem Adressbus einstellen</li> <li>2. Eingabeleitung auf Steuerbus aktivieren</li> <li>3. Daten vom Datenbus (in Register) übernehmen</li> </ol>
Registern verändern	<ol style="list-style-type: none"> <li>1. Rechenlogik aktivieren</li> <li>2. Ergebnis in Register speichern</li> <li>3. Befehls(adress)zähler erhöhen</li> </ol>
2 Registern verknüpfen	<ol style="list-style-type: none"> <li>1. Rechenlogik aktivieren</li> <li>2. Ergebnis in ein Register speichern</li> <li>3. Befehls(adress)zähler erhöhen</li> </ol>
Registern auswerten	<ol style="list-style-type: none"> <li>1. Rechenlogik aktivieren</li> <li>2. je nach Ergebnis den Befehls(adress)zähler verändern</li> </ol>
Befehlszähler verändern	<ol style="list-style-type: none"> <li>1. Befehlszähler einstellen</li> </ol>
Daten aus Register speichern	<ol style="list-style-type: none"> <li>1. Adresse auf Adressbus einstellen</li> <li>2. Daten auf den Datenbus legen</li> <li>3. Schreibleitung auf Steuerbus aktivieren</li> </ol>
Daten aus Speicher lesen	<ol style="list-style-type: none"> <li>1. Adresse auf Adressbus einstellen</li> <li>2. Leseleitung auf Steuerbus aktivieren</li> <li>3. Daten vom Datenbus übernehmen</li> </ol>
Daten ausgeben	<ol style="list-style-type: none"> <li>1. Adresse auf Adressbus einstellen</li> <li>2. Daten auf den Datenbus legen</li> <li>3. Schreibleitung auf Steuerbus aktivieren</li> </ol>



## Vergleich und Gegenüberstellung von VON-NEUMANN- und Harvard-Architektur

	VON-NEUMANN-Architektur	Harvard-Architektur
<b>Gemeinsamkeiten</b>	<ul style="list-style-type: none"> <li>• Bauelemente: Busse, CPU, Speicher, I/O-Systeme</li> <li>•</li> </ul>	
<b>Unterschiede</b>	<ul style="list-style-type: none"> <li>• gemeinsamer Haupt-Bus (geteilt in Daten-, Adress- und Steuer-Bus)</li> <li>• zentraler Bus</li> <li>• periphere CPU</li> </ul>	<ul style="list-style-type: none"> <li>• getrennte Busse für Daten und Befehle</li> <li>• Busse peripher angelegt</li> <li>• zentrale CPU</li> </ul>
<b>Vorteile</b>	<ul style="list-style-type: none"> <li>• nur ein Bus, der universell genutzt werden kann</li> <li>• Speicher ist flexibel zwischen Daten und Programmen aufteilbar</li> </ul>	<ul style="list-style-type: none"> <li>• Daten und Befehle können gleichzeitig, aber auch unabhängig voneinander geladen werden</li> <li>• da Trennung von Daten und Programm in unterschiedlichen Speichern, kann fehlerhafter od. manipulierter Code keine Programme überschreiben (nur Daten)</li> </ul>
<b>Nachteile</b>	<ul style="list-style-type: none"> <li>• Bus mit seiner Geschwindigkeit ist der Flaschenhals des Systems</li> <li>• Daten und Programme nicht unterscheidbar</li> </ul>	<ul style="list-style-type: none"> <li>• freier Speicher lässt sich nicht für die jeweils anderen Daten bzw. Programme nutzen</li> </ul>

---

## **Biographie: John VON NEUMANN (1903 - 1957)**

Geburtsname János Lajos Neumann von Margitta

Rechner-Architektur  
Spiel-Theorie

### 1.2.3. Prozesse

#### Definition(en): Prozess

Ein Prozess ist ein Programm oder ein Programm-Teil, der von einem Prozessor als eine zusammenhängende Einheit betrachtet wird.

Ein Prozess ist ein Programm oder ein Programm-Teil, der vom Betriebssystem als eigenständige Einheit verstanden wird und vom Prozessor oder einem Teil (Kern) abgearbeitet wird.

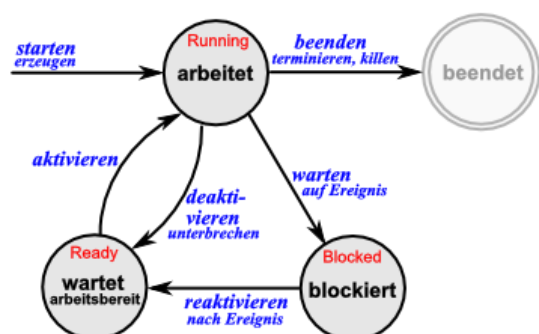
Ein Prozess ist eine von einem Betriebssystem gesteuerte, eigenständige Informations-Verarbeitungs-Einheit.

Ein Prozess ist die konkrete Instanzierung eines Programm's zu dessen Abarbeitung im Informations-Verarbeitungs-System.

Ein Prozess ist die Ablauf-Umgebung eines Programm's in einem Rechner-System.

#### Prozess-Zustände (aus der Sicht des Betriebssystems)

- arbeitet
- wartet
- blockiert



Betriebs-System hat Prozess-Scheduler, der die verschiedenen Prozesse, die eigentlich gleichzeitig ablaufen müssten, verwaltet

Möglichkeit wäre Liste von Prozessen mit einer Reihenfolge entsprechend der Priorität  
Gefahr, dass Prozesse mit geringer Priorität ganz untergehen

mehrere Prozessoren oder Prozessoren mit mehreren Kernen können mehrere unabhängige Prozesse parallel / nebenläufig ausführen

---

gleiches gilt für mehrere Instanzen des gleichen Programms  
nebenläufige Prozesse können nur auf globale Ressourcen (z.B. Maus-Position, Tastatur-Eingaben, ...) zugreifen, meist ist nur ein lesender Zugriff erlaubt  
das Schreiben setzt i.A. exklusive Prozesse voraus, sie greifen nur alleine (eben exklusiv) auf eine Ressource zu

mit bestimmten Befehlen kann man die exklusive Abarbeitung eines Programm-Abschnitts realisieren  
in dieser Zeit kann dann kein Interrupt oder ein anderer Prozess die Abarbeitung des Programm-Abschnitts verhindern

besonders ausgenutzt wird die Nebenläufigkeit von den Graphik-Prozessoren auf Graphik-Karten. Sie brechnen in vielen Kernen gleichzeitig mehrere Graphik-Elemente (meist Pixel).  
die Graphik-Prozessoren sind häufig speziell optimierte RISC-Prozessoren

Semaphoren  
sind Markierungen / Anzeiger für bestimmte System-Zustände oder Verfügbarkeiten von Ressourcen

besser Zeitscheiben  
Round-Robin-Scheduler  
Prozesse bekommen Anteile an Prozessor-Zeit  
relativ gerechte Verteilung der Ressourcen  
Nachteil ist, dass auch hoch-priorisierte Prozesse länger brauchen  
am Ende der zugewiesenen Prozessorzeit muss Zustand des Prozessor's, des Stack's und ev. andere Speicherbereiche (z.B. Cache) usw. gesichert werden muss und dann natürlich vor dem Vortsetzen im nächsten Zeit-Abschnitt wieder geladen werden müssen  
das kostet Zeit  
würde die Sicherung der Prozess-Zustände nicht passieren und Prioritäten nicht beachtet werden, dann könnte z.B. ein heilloses Daten-Chaos entstehen. Mehrere Druck-Prozesse können z.B. nicht gleichzeitig auf einen Druck zugreifen. Das würde dann wohl gemischte Ausdrücke ergeben.  
Ein Drucker ist also z.B. eine kritische oder auch exklusive Ressource. Nur ein Prozess kann / darf zu einer Zeit auf sie zugreifen. Andere Prozesse müssen dann in den "blockiert"-Zustand versetzt werden, auch wenn ihnen gerade CPU-Zeit zugewiesen wurde.

Andere Prozesse können nebenläufig abgearbeitet werden. So behindern sich das Drucken eines Dokumentes und die Anzeige einer Maus-Bewegung praktisch nicht.

Betriebssystem müssen Konflikte zwischen Prozessen verhindern / erkennen und die Abläufe u.U. passend steuern

in Windows-Systemen ist ein besonderer niedrig-priorisierter Prozess – der "Leerlaufprozess" – ein Puffer für nicht genutzte Prozessor-Zeit

heutige Rechner mit ihren Betriebssystemen sind Kombinationen aus mehreren Kernen in einer CPU und optimierter Zeitscheiben-Steuerungen.

Interrupt's

---

Um das Eintreffen bestimmte Ereignisse zu beobachten, müssten viele Port's und Schnittstellen überwacht werden. Das würde sehr viel Arbeits-Zeit kosten. Hier hat man sich bei modernen Rechnern für einen anderen Mechanismus entschieden. Bestimmte Ereignisse können Unterbrechungen – engl. Interrupts – auslösen. Dabei wird die normale Abarbeitung der Prozesse unterbrochen, der Prozess-Zustand gesichert und dann mit der Interrupt-Routine auf das Unterbrechungs-Ereignis reagiert. Danach wird dann wieder mit der normalen Prozess-Bearbeitung weiter gemacht.

### **Interrupt-Arten**

- **Hardware-Interrupt** werden durch technische Bestandteile ausgelöst
- **Software-Interrupt** werden von einem Programm ausgelöst

Hardware-Interrupt-Routinen waren in älteren Betriebssystem oft der Einstiegs-Punkt für Viren und ähnliche System-Schädlinge  
heute hat der Nutzer keinen direkten Zugriff mehr auf die Hardware, die Interrupts und deren Behandlungs-Routinen. Sie sind vom restlichen System abgekapselt (→ Schalen-Modell).

### **Interrupt-Arten**

- **System-Interrupt** Interrupts, die sich aus bestimmten System-Bedingungen ergeben und eine Reaktion des Betriebssystem erfordern
- **Anwendungs-Interrupt** Unterbrechungen, die sich aus der Abarbeitung eines Programm's / Algorithmus ergeben, wie z.B. Division durch Null  
Anwendung reagiert auf einen Abarbeitungs-Fehler
- 

<b>Definition(en): Interrupt</b>
Ein Interrupt ist eine Unterbrechung der normalen Prozess-Bearbeitung, um auf ein Ereignis mit einem speziellen Programm (Interrupt-Routine) zu reagieren.

---

**Definition(en): Task**

Ein Task ist eine Aufgabe (ein Programm / Service), die vom (Betriebs-)System ausgeführt wird.

Ein Task kann aus mehreren Prozessen bestehen.

Thread (dt.: Faden)

eigenständiger / abgegrenzter Ausführungs- od. Programm-Strang innerhalb eines Prozesses

**Definition(en): Thread**

Ein Thread ist ein (meist kleiner) Programm-Abschnitt (Algorithmen-Teil), der zusammenhängend abgearbeitet werden soll, auf ein bestimmtes Daten-Segment zugreift und ev. über bestimmten Referenzen / Zeiger auf das Daten-Segment verfügt.

User-Thread ist ein entsprechender Software-Abschnitt des Anwender-Programms  
im Allgemeinen ist hier keine Steuerung des Betriebssystems notwendig  
Anwendung muss aber ev. bei nebenläufigen Thread's dafür sorgen, dass Abhängigkeiten bei Daten beachtet werden  
ein Thread kann nicht auf einen berechneten Wert eines anderen Thread's zugreifen, wenn diese Berechnung noch nicht abgeschlossen ist

---

## 1.2.4. Programmierung des Rechners / der CPU

eigentlich in dualer Form (Maschinencode)  
praktisch eindeutige Zuordnung von Speicherzelle (Adresse) und ihrem digitalem Inhalt  
(entweder CPU-Befehl oder Daten (z.B. für Register oder Speicher))  
Daten in chiffrierter Form

besser lesbar in Assembler-Sprache geschrieben  
Umschreibungen der CPU-Anweisungen in einfache Mnemonics

Quelltext muss dann in duale Form übersetzt (compiliert bzw. interpretiert) werden (vom sogenannten Assembler:  
von Hand in dualer Form (Maschinencode) oder per Assembler geschriebene Programme  
gleich gross, da eine 1 : 1-Übersetzung erfolgt  
Quelltext durch Mnemonics und Hilfsstrukturen (z.B. Sprungmarken) sowie Kommentare  
deutlich aufgebläht

noch angenehmer für Menschen Programmierung in höherer Programmiersprache:

auch diese Quelltexte müssen in duale Form übersetzt werden  
spezieller Interpreter oder Compiler notwendig  
Programme üblicherweise deutlich größer / länger und damit langsamer als Maschinencode  
Quelltexte lassen sich vielfach für andere CPU's nutzen (allerdings andere Übersetzer benötigt)  
Quelltexte üblicherweise durch sinnvolle Zusammenfassung von Maschinenbefehlen zu höheren Befehlsgruppen (Schlüsselwörter der Programmiersprache) kleiner als Maschinencode

Entwicklung der höheren Programmiersprachen

Grace HOPPER

Crazy Grace, Miss COBOL

Tauschen von MOPS und Johnny  
Betonung der beobachteten Simulation (z.B. in Zeitlupe)

---

### **1.2.4.2. Simulation eines VON-NEUMANN-Rechner mit "Johnny"**

"Johnny" ist ein Simulations-Programm für einen vereinfachten VON-NEUMANN-Rechner. Das von Peter DAUSCHER entwickelte Programm steht unter der GPL V.3 und ist als portable App nutzbar.

Für Windows steht ein Download der ausführbaren Dateien zur Verfügung (→ ). Für andere Betriebssysteme muss der – frei verfügbare Quellcode – kompiliert werden. Die letzte - derzeit ladbare Version 1.01 ist aus dem Jahr 2014.

Im Io-Stick von Timo Hempel (→ <https://tinohempel.de/info/info/IoStick/index.html>) ist "Johnny" fertig integriert und sofort nutzbar.

Das nach dem Download zu entzippende Verzeichnis kann aber auch direkt irgendwohin kopiert werden und daraus die Johnny.EXE gestartet werden. Kopiert man den Ordner in den PortableApps-Ordner eines portable-Apps-Sticks, dann findet man "Johnny" nach dem nächsten Start im Menü.

Da verschiedene Elemente eines VON-NEUMANN-Systems anders abgebildet oder weggelassen wurden, sollte man den Simulator wirklich nur als Simulator verstehen, nicht als Darstellung des Modell's. Auch Tatsachen (Speicher-Inhalte) und Interpretationen (z.B. Asm + Opnd) sind teilweise gemischt.

Im Folgenden werden wir uns vorrangig mit der Bedienung beschäftigen, da diese dann für die Übungs-Aufgaben gebraucht wird.

#### **Links:**

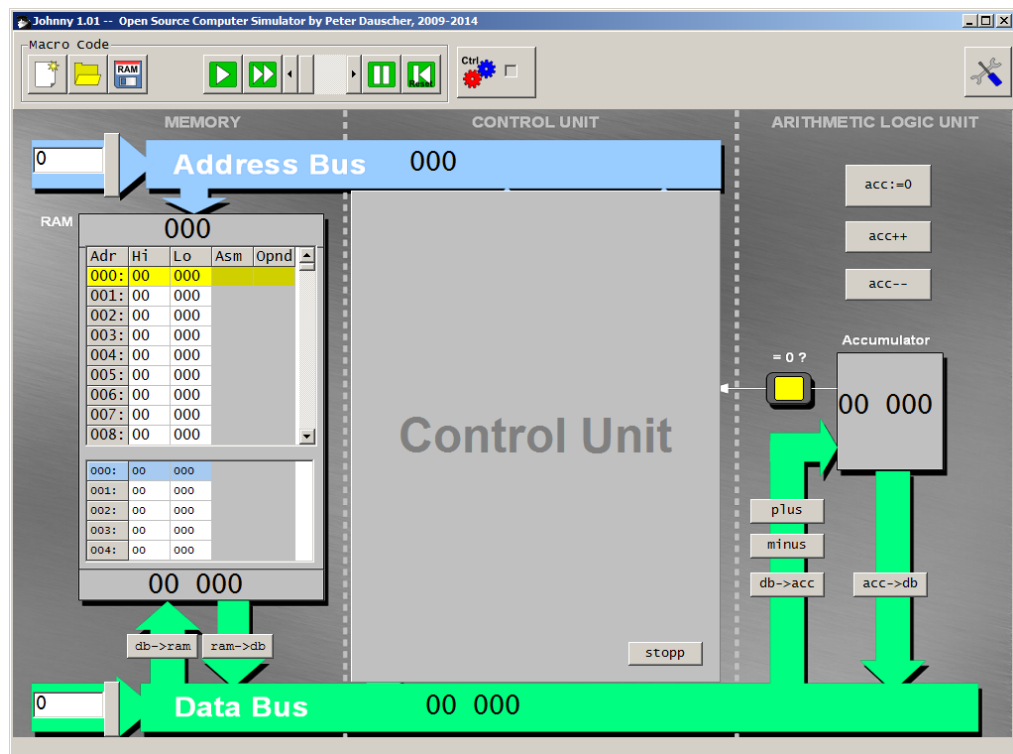
<https://www.youtube.com/watch?v=74UbqN4o9Po> (sehr kleine Einführung; P. SCHNABEL; 02:11)

[https://wiki.zum.de/wiki/Rechnerarchitektur\\_mit\\_Simulator\\_JOHNNY](https://wiki.zum.de/wiki/Rechnerarchitektur_mit_Simulator_JOHNNY) (Wiki + Info; P. DAUSCHER) [5★]



### 1.2.2.2.1. Aufbau des Simulators

Nach dem Start und dem Weg-Klicken der Programm-Info (Splash-Screen) sehen wir vier Elemente eines VON-NEUMANN-Rechner's.



Dazu gehören der Speicher (Memory) und der Prozessor mit Kontroll-Einheit und der Arithmetik-Logik-Einheit sowie das Bus-System. Das Bus-System besteht hier nur aus Adress- und Daten-Bus.

Da auf die Eingabe-Ausgabe-Einheiten (IO-Unit's) verzichtet wurde, benötigen wir auch nicht unbedingt einen Steuer-Bus.

Praktisch kann man sich den Simulator als Prototypen eines sehr einfachen Prozessor's vorstellen, der auf seinen eigenen Speicher (Cache) zugreift.

Mittels Symbol-Leiste über dem abgebildeten Prozessor finden wir die Bedien-Elemente des Simulator's sowie die Einstellungen.

### Speicher(-Werk) (Memory)

Der Speicher (RAM ... Random Access Memory → beliebiger Zugriffs-Speicher) hat 1'000 Speicherzellen, die über die Adressen 000 bis 999 angesprochen werden können. Die Auswahl einer Speicherzelle wird über den Adress-Bus gesteuert.

Die Anzeige des Speicher's ist in zwei Bereiche geteilt. Der obere Bereich dient als Eingabe-Bereich. Hier lassen sich Daten und Programme ablegen. Bei VON-NEUMANN-Rechner liegen beide ja gemeinsam im gleichen Speicher.

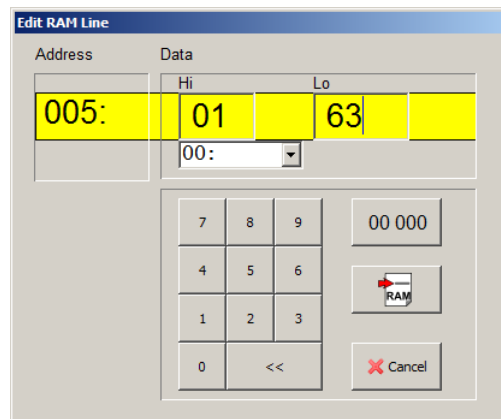
Der untere Bereich dient der Anzeige der aktuellen Aktivitäten (Lesen-Speichern) im RAM. So kann dann später im oberen Bereich das Lesen der Programm-Anweisungen und unten das Lesen und Schreiben der Daten simultan beobachtet werden.

RAM				
005				
Adr	Hi	Lo	Asm	Opnd
000:	19	999		
001:	00	000		
002:	00	000		
003:	00	000		
004:	00	000		
005:	00	022		
006:	00	000		
007:	00	000		
008:	00	000		
009:	00	000		

00 022

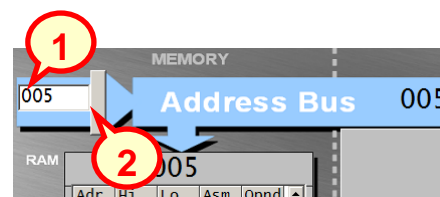
Die Speicher-Zellen lassen sich durch Klicken auf die Anzeige-Zeile einer Speicherzelle im oberen Bereich mit Werten belegen.

Als Werte sind hier Zahlen zwischen 0 und 19'000 erlaubt. Der Gesamt-Wert wird in den High-Wert und den Low-Wert zerlegt. Der High-Wert sind die Tausender und Zehntausender. Die Einer bis Hunderter kommen in den Low-Bereich. In echten Systemen haben wir ja auch ein High-Byte und ein Low-Byte in einem 16-bit-System. Ein solches direktes Hinein-Schreiben in den Speicher ist natürlich in echten Systemen nicht möglich. Einzig die Benutzung eines ROM (Read Only Memory → Nur-Lese-Speicher) wäre so simulierbar (entspricht dem Brennen eines ROM).



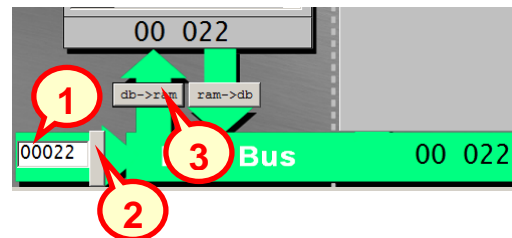
Der normale Weg wäre über die Bus-Systeme möglich. Dabei könnten die Daten von irgendwelchen IO-Geräten stammen.

Zuerst müssen wir die zu benutzende Speicherzelle mit ihrer Adresse spezifizieren. Dazu wird zuerst die (dezimale) Adresse angegeben und diese dann über die schmale (unbeschriftete) Schaltfläche auf den Daten-Bus geschrieben.



Hat alle geklappt, erscheint die Adresse oben in der Speicher-Anzeige.

Mit dem Inhalt gehen wir sachlich ähnlich vor. Nur muss am Schluß der Wert vom Daten-Bus noch in den Speicher geschrieben werden. Dazu dient die Schaltfläche "db → ram".



Sachlich könnten alle am Bus angeschlossenen Geräte / Einheiten den Daten-Bus auslesen. In der Praxis wird die Berechtigung dazu über den Steuer-Bus geregelt.

Die Speicher-Belegung (bisher sind es ja nur Daten) lassen sich als ram-Datei in Johnny speichern. Später kann die Datei dann auch das zugehörige Programm enthalten – es ist ja auch nur eine Speicher-Belegung im RAM.

### Aufgaben:

1. **Belegen Sie den RAM auf den Speicherzellen 000 und 001 mit den Zahlen 13 und 4! Speichern Sie sich die Speicher-Belegung als Übung1.ram ab!**
2. **Belegen Sie nun den Speicher beginnend bei 004 mit den ersten 5 ungeraden natürlichen Zahlen! Speichern Sie sich die Speicher-Belegung als Übung2.ram ab.**
- 3.

## Rechenwerk (ALU ... Arithmetic Logic Unit → Arithmetik-Logik-Einheit)

Die ALU ist das Kernstück moderner Prozessoren und dort mit Hunderten Funktionen ausgestattet. In unserem Simulator sind es nur sehr wenige.

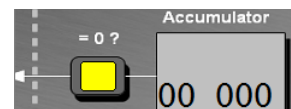
Um mit verschiedenen Zahlen – oder besser Bit-Mustern arbeiten zu können – verfügen die ALU's über eine Vielzahl interner Speicher-Plätze. Diese werden Register genannt. Das Haupt-Register heißt Akkumulator. In diesem werden die eigentlichen Funktionen ausgeführt.

Unser Johnny-Computer kann in seiner ALU die folgenden Funktionen realisieren:

Funktion	Schaltfläche	
<b>addiere</b> Akkumulator und Daten-Bus	<b>plus</b>	Akkumulator := Akkumulator + Daten-Bus
<b>subtrahiere</b> Daten-Bus vom Akkumulator	<b>minus</b>	Akkumulator := Akkumulator - Daten-Bus
<b>lade</b> Daten-Bus in Akkumulator	<b>db-&gt;acc</b>	Akkumulator := Daten-Bus
<b>gebe</b> Akkumulator auf Daten-Bus <b>aus</b>	<b>acc-&gt;db</b>	Daten-Bus := Akkumulator
<b>setze</b> Akkumulator <b>auf 0</b>	<b>acc:=0</b>	Akkumulator := 0
<b>inkrementiere</b> den Akkumulator	<b>acc++</b>	Akkumulator := Akkumulator + 1
<b>dekrementiere</b> den Akkumulator	<b>acc--</b>	Akkumulator := Akkumulator - 1

Nebenbei prüft die ALU ständig, ob der Akkumulator den Wert Null hat. Ist das so, dann wird ein Signal (Flag) gesetzt.

In Johnny ist das durch das Leuchten des gelben Lämpchen zu erkennen.



All diese Funktionen werden in einer echten ALU natürlich nicht auf Knopfdruck oder durch irgendwelche Männchen erledigt. Das Auslösen einer Funktion erfolgt Programm-gesteuert durch die Controll-Einheit. Dazu später mehr.

Wollen wir nun z.B. zwei Zahlen addieren, die in den Speicherzellen 000 und 001 stehen, dann brauchen wir die folgenden Arbeitsschritte.

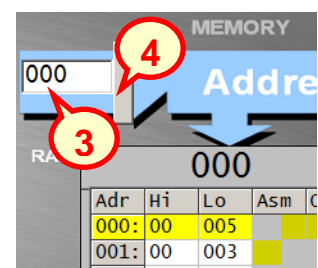
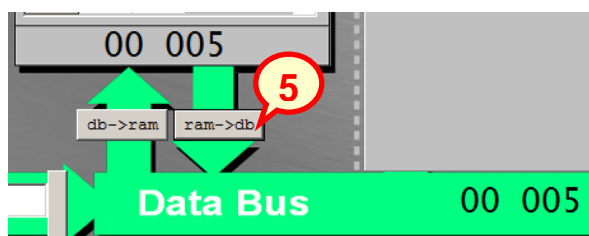
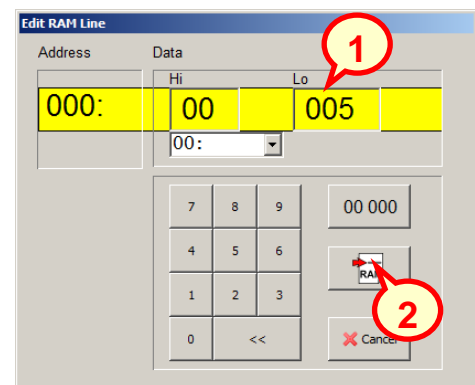
Zuerst wählen wir die passende Speicherzelle durch Klicken aus. Im folgenden Dialog können wir nun Hi- und Low-Wert der Zelle im gelben Bereich ändern (1). Die Änderung wird dann wirksam, wenn die RAM-Taste gedrückt wurde (2).

Genau so verfahren wir mit der zweiten Speicherzelle. Hier könnte z.B. eine 3 eingegeben werden.

Die Rechen-Operationen können nur in der ALU erfolgen. Also müssen die Daten dahin transportiert werden.

Zuerst wählt man die passende Adresse auf dem Adress-Bus (3 + 4).

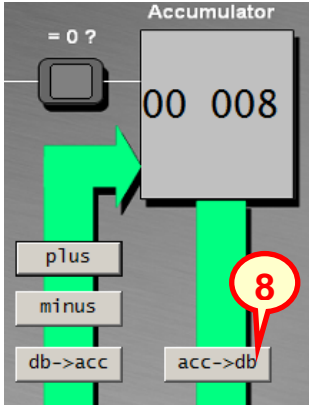
Nun kann mit [ram -> db] der Inhalt der gewählten Speicherzelle auf den Daten-Bus gebracht werden (5).



Die ALU kann nun den Daten-Bus lesen und den Inhalt in den Akkumulator (das Haupt-rechen-Register) übernehmen (6).

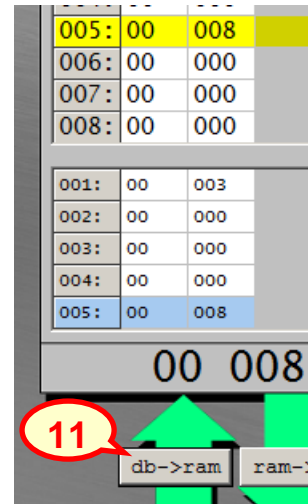
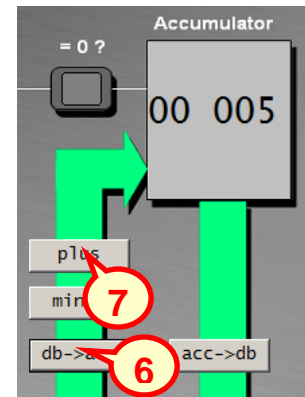
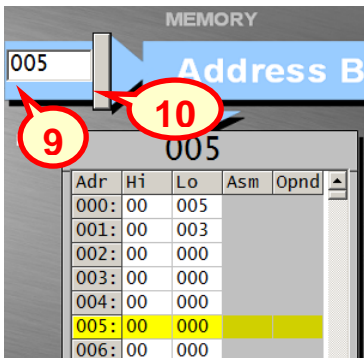
Als nächstes muss der zweite Operant auf den Daten-Bus gebracht werden. Dazu wiederholen wir die die Schritte 3 bis 5 mit der Speicherzelle 001.

Die ALU kann nun den Daten-Bus-Inhalt zum Akkumulator dazuzaddieren ([plus]; (7)) oder abziehen ([minus]).



Nachdem der Akkumulator nun das Ergebnis enthält, muss dieses in den Speicher zurück. Der Weg geht über den Daten-Bus (8) in die passende Speicherzelle. Diese muss aber erst adressiert werden. Die gewünschte Zelle soll die Adresse 005 haben. Das Vorgehen haben wir schon besprochen. Zuerst wird die Adresse (9) auf dem Adress-Bus eingestellt (10) und dann noch der Transfer vom Daten-Bus ausgelöst (11).

Damit ist die Aufgabe erledigt.



---

### Aufgaben:

1. **Geben Sie in die Speicherzellen 010, 011 und 012 die Zahlen 24, 4 und 13 ein! Überlegen Sie sich die Schritte, die gemacht werden müssen, um die drei Zahlen zu addieren und das Ergebnis in der Zelle 004 zu speichern! Probieren Sie die Arbeitsschritte immer einzeln aus! Notieren Sie die gemachten Schritte und machen Sie sich daneben Bemerkungen, was jeweils bei dem Schritt gemacht wird! Speichern Sie sich die Speicher-Belegung als ram-Datei ab!**
2. **Erstellen Sie ein Programm dass die Inhalte der Speicherzellen 011 und 012 von der Zelle 004 subtrahiert! Das Ergebnis soll nun in der Speicherzelle 009 gespeichert werden! Notieren Sie sich die Befehle! Speichern Sie sich den Speicher-Inhalt ab!**
3. **Laden Sie die Speicher-Belegung Übung2 und addieren Sie die Zahlen in die Zelle 002!**
4. **Erstellen Sie die folgende Speicherbelegung!**
  - a) **fortlaufend ab der Speicherzelle 030 die geraden Zahlen beginnend bei 2 bis einschließlich 10**
  - b) **beginnend bei der Speicherzelle 995 rückwärts die Zahlen 1 bis 20**
5. **Geben Sie in den Adressen 010 und 011 die Zahlen 1'001 und 2'002 ein und addieren Sie beide über die ALU! Das Ergebnis soll dann in der Speicherzelle 012 abgelegt werden! Notieren Sie sich alle Arbeitsschritte!**
- 6.

Simulation von Programmen

Beobachten und Protokollieren zum Programm-Ablauf (z.B. Befehl-Zähler, bestimmte (Daten-)Adressen)

Verändern der Daten im Programm

Verändern der Funktion des Programm's

Entwicklung eigener Programme

Kartenspiel "Digitalo" (Inf-Schule)

Video's Logische Operationen / schalter mit Domino-Steine

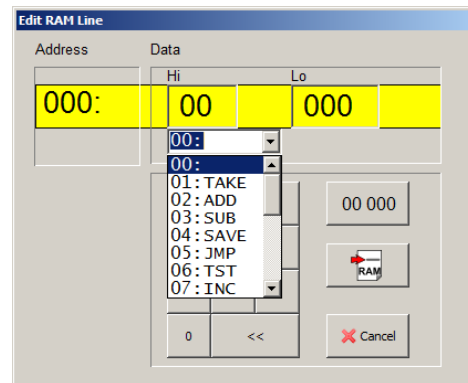
Flip-Flop und Halb-Addierer zusammenstellen / entwickeln

## Maschinen- und Assembler-Programmierung

Aus den einzelnen Funktionen sinnvolle größere Daten-Operation zusammensetzen ist schon beachtlich aufwendig. Da bestimmte Aufgaben, wie z.B. das Laden des Inhaltes aus einer bestimmten Speicherzelle immer wieder die gleichen Einzel-Operationen erfordern, sind solche Aufgaben als feste Befehle im System gespeichert.

Es sind praktisch die Maschinencode-Befehle, die im Speicher als Programm liegen. In Johnny werden diese Makro's genannt. Die Maschinencode-Befehle sind immer im High-Adressteil anzugeben. Der Low-Teil enthält dann eventuell notwendige Operanden-Daten. Bei der Eingabe sehen wir neben dem Maschinencode – bei Johnny sind das die Befehle 00 bis 19 – auch gleich eine Text-Angabe. Hierbei handelt es sich um die Menschen-lesbare Form – dem Assemblercode.

Die Befehle 11 bis 19 kann man selbst definieren.



Maschinen-Befehl	Assembler- / Makro-Befehl	Operand	Beschreibung / Funktion
00			
01	TAKE	Adresse	der Wert der angegebenen absoluten <b>Adresse</b> wird in den Akkumulator <b>geladen</b>
02	ADD	Adresse	der Wert der angegebenen absoluten <b>Adresse</b> wird zum Akkumulator dazu <b>addiert</b>
03	SUB	Adresse	der Wert der angegebenen absoluten <b>Adresse</b> wird vom Akkumulator <b>subtrahiert</b>
04	SAVE	Adresse	der Wert des Akkumulator wird in der angegebenen absoluten <b>Adresse gespeichert</b>
05	JMP	Adresse	<b>Programm-Fortsetzung</b> an der angegebenen <b>Adresse</b> (Programm-Zähler setzen))
06	TST	Adresse	testet den Wert der angegebenen (Daten-) <b>Adresse</b> ; wenn der <b>Vergleich 0</b> ergibt, wird die nachfolgende (Programm-)Adresse übersprungen (die direkt nachfolgende (Programm-)Adresse kann dann z.B. für HLT oder JMP genutzt werden (→ Schleifen-Konstrukt))
07	INC	Adresse	<b>erhöht</b> den Wert in der angegebenen absoluten <b>Adresse um 1</b>
08	DEC	Adresse	<b>erniedrigt / verringert</b> den Wert in der angegebenen absoluten <b>Adresse um 1</b>
09	NULL	Adresse	<b>setzt</b> den Wert der angegebenen <b>Adresse auf 0</b>
10	HLT	---	Halt bzw. Stopp (Programm-Ende) → Johnny erzeugt eine Bildschirm-Meldung
11 - 19			frei programmierbare Makro's bzw. eigene Assembler-Anweisungen

Die programmier-technische Seite schauen wir uns gleich an (→ ). Zuerst sollen noch die Elemente von Johnny vollständig besprochen werden.

Auf die Kompatibilität zu bzw. die Nutzung des Bonsai-Modus vernachlässigen wir hier. Wer dahingehend interessiert ist oder mit diesem Modus arbeiten will / muss, dem sei die Bedienungs-Anleitung angeraten. Mehr noch wird man sich auf die Suche nach Sekundär-Literatur machen müssen, da die Beschreibung recht knapp ausfällt.

---

**Links:**

<https://bonsai.pinyto.de/> (Bonsai-Rechner online)

## Kontroll-Einheit (Control Unit → Steuerungs-Einheit)

In der Kontroll-Einheit werden die einzelnen Maschinen-Befehle ausgeführt. Dabei werden sie in die notwendigen Mikro-Code-Befehle für einen Durchlauf des VON-NEUMANN-Zyklus zerlegt und dann getaktet ausgeführt.

Normalerweise ist die Control-Einheit verdeckt, da die Abläufe ohne Vorkenntnisse hier schwer zu verfolgen sind. Mit dem "Ctrl-Zahnräder"-Button kann die Control-Einheit aufgedeckt werden. Die Symbol- bzw. Menü-Leiste erweitert sich dann auch um Steuer-Elemente für den Umgang mit Mikro-Befehlen.

Drei Elemente sind für die interne Abarbeitung von Programmen in der CPU bedeutsam. Da ist zum Ersten der Programm-Zähler (Prg.Counter). Er enthält die Adresse des abzuarbeitenden Befehls im Speicher. Der Begriff des Zählers ist dabei etwas irreführend. Vielmehr müsste es Befehls-Adressen-Register od. so ähnlich heißen.

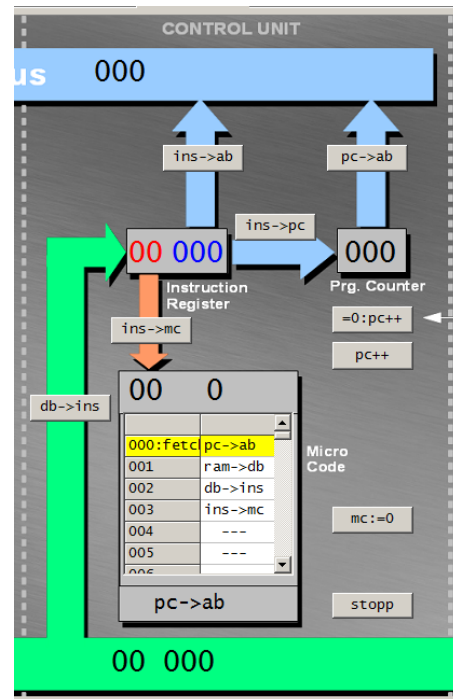
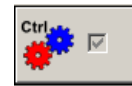
Das Befehls-Register (Instruction Register) enthält den aktuellen Maschinen-Befehl. Der vordere Teil (**rot**) wird für die innere Mikrocode-Abarbeitung benötigt. Der rechte Teil (**blau**) steht für eine Adresse. In echten Prozessoren kann dieser Teil aber auch andere Operanden enthalten. So etwas ist in Johnny nicht umgesetzt.

Die Adresse im Instruktionen-Register kann für die direkte Steuerung des Adress-Busses genutzt werden. Fast alle Assembler-Befehle in Johnny haben Adressen als Operand.

Weiterhin kann der Befehlszähler neu eingestellt werden. Das passiert eben bei Sprung-Befehlen.

Die Anzeige der Micro-Code-Sequenzen ermöglicht uns einen Einblick in die Einzelschritte einer Befehls-Abarbeitung.

Die Abarbeitung des VON-NEUMANN-Zyklus für einzelne Assembler-Befehle sehen wir uns etwas weiter hinten an (→ [1.2.2.2.2. Beobachtung des VON-NEUMANN-Befehls-Zyklus](#)).



### Aufgaben:

- 1.
- 2.
- 3.
4. *Erstellen Sie nun ein Programm (Makro), dass ... !*



## 1.2.2.2. Programmierung von Johnny

Mit Hilfe der Assembler-Befehle (Makro's) können wir nun Programme schreiben, die sehr stark an Assembler-Codes für moderne CPU's erinnern. Praktisch lässt sich auf dieser Ebene alles programmieren, was ein moderner Computer kann. Man wird aber schnell feststellen, dass Aufwand und Nutzen in keinem vernünftigen Verhältnis stehen. Da müssen bessere Programmier-Tools her, um die heutigen Probleme zu lösen. Zu solchen Tools gehören z.B. höhere Programmiersprachen wie BASIC, Python, JAVA usw. usf. (s.a. → [📖 Sprachen und Automaten](#), [📖 Python](#), [📖 JAVA](#))

Daneben sind es Datenbank-Management-Systeme, die einen etwas anders gearteten Zugang zu den Computer-Systemen von heute zulassen (s.a. → [📖 Datenbanken](#)).

### 1.2.2.2.1. Erstellen einfacher Assembler-Programme

Programme in Johnny sollten mit der Adresse 000 beginnen. Darauf sind verschiedene Bedien-Elemente von Johnny eingerichtet. Wer will kann natürlich auch ab irgendeiner anderen Adresse starten.

Schauen wir uns zuerst ein fertiges Programm an, um die Teile des Assembler-Code's und die "Denkweise" von Johnny kennen zu lernen.

Addieren zweier Zahlen			
Speicher-Adresse	Assembler-Befehl (Asm)	Operand (Opnd)	Kommentare / Beschreibungen / Hinweise
000	TAKE	010	lade den Inhalt von Adresse 010 in den Akkumulator
001	ADD	011	addiere den Inhalt von Adresse 011 zum Akkumulator
002	SAVE	012	speichere den Akkumulator in der Adresse 012
003	HLT	000	beende das Programm
004	00	000	<i>nicht relevante Zell-Inhalte sind ausgegraut</i>
010	00	004	1. Datum
011	00	003	2. Datum
012	00	000	Ergebnis-Adresse
013	00	000	

#### **Aufgaben:**

- 1. Erstellen Sie das Additions-Programm im Speicher!**
- 2. Sichern Sie den Speicher (RAM) mittels "Save RAM as Johnny File" ("Speichern unter ...")!**
- 3. Starten Sie das Programm mit dem Abspiel-Button! (Achten Sie darauf, dass Sie mit Adresse 000 starten!)**
- 4. Verfolgen Sie nun schrittweise die Abarbeitung Ihres Programms!**

Ein gutes Mittel, die Programm-Abläufe zu verfolgen sind Protokolle. Wir dokumentieren dabei nur die Veränderungen. Steht kein neuer Wert im Protokoll, dann gilt der letzte notierte Inhalt. (Im nächsten Beispiel verwenden wir eine Ausgrauung! Später lassen wir diese Inhalte komplett weg.)

Takt	Speicher		Busse		CPU			
	akt. Adr.	Adr.-Inhalt	Adr.-Bus	Daten-Bus	Inst.-Reg.	Akk.	Prog.-Zhr.	
0	000	01'010	000	00'000	00'000	00'000	000	
1	001	02'011	010	00'004	01'010	00'004	001	
2	002	04'012	011	00'003	02'011	00'007	002	
3	003	10'000	012	00'007	04'012	00'007	003	
4	<b>STOP</b>	10'000	003	10'000	10'000	00'007	003	

Wir merken hier schnell, dass sich hier viele Informationen doppeln, so dass wir auf eine Protokollierung ev. verzichten können. Das werden wir zukünftig nach Bedarf tun. Was uns aber auch interessiert, sind die bearbeiteten Speicher-Adressen unserer Daten. Da diese sich im Laufe der Programm-Arbeitung ebenfalls ändern können, nehmen wir sie in die Spalten-Reihe mit auf.

Ein gut nachvollziehbares Protokoll sieht dann z.B. so aus.

Takt	Speicher		Busse		CPU			Daten-RAM		
	akt. Adr.	Adr.-Inhalt	Adr.-Bus	Daten-Bus	Inst.-Reg.	Akk.	Prog.-Zhr.	010	011	012
0	000	01'010	000	00'000	00'000	00'000	000	004	003	000
1	001	02'011	010	00'004	01'010	00'004	001	004	003	000
2	002	04'012	011	00'003	02'011	00'007	002	004	003	000
3	003	10'000	012	00'007	04'012	00'007	003	004	003	007
4	<b>STOP</b>	10'000	003	10'000	10'000	00'007	003	004	003	007

Für einfache Programme scheint der Aufwand etwas groß. Aber spätestens bei komplexeren Programmen mit Vergleichen, Sprüngen usw. wird dieses Vorgehen bei der Fehler-Analyse nicht mehr zu umgehen sein. Früh übt sich, wer ein Meister werden will. Auch die besten Musiker haben mit dem Üben der Ton-Leiter angefangen.

### Aufgaben:

1. Ändern Sie das Programm so ab, dass die Zelle 010 von der Zelle 011 abgezogen wird! Im RAM steht unter der Adresse 010 eine 30 und in 011 eine 45.
2. Notieren Sie das Programm zuerst auf Papier und gehen Sie es theoretisch durch (Papier-Simulation)! (Sie dürfen sich natürlich ein "Protokoll" mit-schreiben.)
3. Geben Sie das Programm ein, speichern Sie es und führen Sie es dann schrittweise aus!
4. Wenn das Programm exakt arbeitet, starten Sie es erneut und erstellen Sie sich ein vollständiges Protokoll!

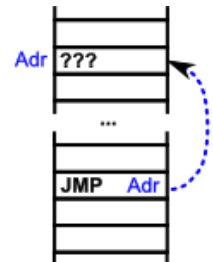
### Übungs-Aufgaben:

1. Erstellen Sie ein Programm, das die Inhalte der Adressen 020, 021 und 022 addiert und unter der Adresse 025 abspeichert!
2. Gesucht ist ein Programm, das die Inhalte der Adressen 030, 031, 032 und 033 vor der Addition um 1 verringert! Das Ergebnis soll unter der Adresse 034 abgespeichert werden!
3. Erstellen Sie ein Papier-Protokoll für den Fall, das Sie das (funktionierende) Programm von Aufgabe 2 erneut ablaufen lassen!
4. Lassen Sie das Programm nun ein zweites Mal ablaufen und vergleichen Sie den Programm-Ablauf mit Ihrem "vorausgesagtem" Papier-Protokoll!
- 5.

### Programme mit unbedingten Wiederholungen / Schleifen / Schlaufen

Eigentlich würden wir Schleifen erst nach den Verzweigungen besprechen, aber bei dem eingeschränkten Befehls-Satz unseres Johnny-Rechner's brauchen wir für viele Aufgaben den Sprung-Befehl JMP.

Seine Wirkungsweise ist schnell erklärt. Als Operand erwartet JMP eine Programm-Adresse, d.h. die Stelle im Programm, an der das Programm weiter fortgesetzt werden soll. Ob die Ziel- bzw. Sprung-Adresse dabei vor oder hinter der aktuellen Befehls-Adresse liegt, ist egal. Praktisch manipuliert der JMP-Befehl den Programmzähler.



Springen wir hinter die aktuelle Programm-Adresse, dann wird zunächst einmal ein Teil des nachfolgenden Programm-Codes ausgelassen. Beim Sprung nach vorne – also zu einer kleineren Adresse (s.a. Abb.) – bewirkt eine unendliche Schleife. Das Programm wird ja wieder von der (kleinen) Sprung-Adresse bis hin zum JMP-Befehl abgearbeitet und dann wieder der Sprung durchgeführt – und das immer wieder und wieder.

Solche Schleifen kann man durch einen Trick auch wieder verlassen – dazu gleich mehr (→ [Programme mit bedingten Schleifen / Wiederholungen / Schlaufen](#)).

Das nachfolgende Programm soll immer abwechselnd eine 11'111 bzw. eine 00'000 in die Speicherzelle 020 schreiben.

Abwechselndes Abspeichern zweier Zahlen			
Speicher-Adresse	Assembler-Befehl (Asm)	Operand (Opnd)	Kommentare / Beschreibungen / Hinweise
000	TAKE	010	lade den Inhalt von Adresse 010 in den Akkumulator
001	SAVE	020	speichern des Akkumulator in die Zelle 020
002	TAKE	011	lade den Inhalt von Adresse 011 in den Akkumulator
003	SAVE	020	speichern des Akkumulator in die Zelle 020
004	JMP	000	springe zum 1. Befehl der Wiederholung
005	00	000	<i>nicht relavante Zell-Inhalte sind ausgegraut</i>
010	11	111	1. Datum
011	00	000	2. Datum
020			Ergebnis-Adresse

**Aufgaben:**

1. Erstellen Sie ein Programm, das die Muster 11'011, 10'101 und 10'001 immer abwechselnd in die Speicherzelle 030 speichert!
2. Lassen Sie ein Programm den Inhalt der Zelle 005 immer jeweils um 1 erhöhen! Beobachten Sie den Verlauf des Programm's!
3. In den Zellen 020 und 021 sind die Zahlen 11'111 einzuspeichern! Mittels eines Programms soll der Wert der Zelle 020 immer um 1 erhöht und der von 021 um 1 verringert werden! Beobachten Sie den Verlauf des Programm's!

**Programme mit Verzweigungen / Entscheidungen / Unterscheidungen**

In der Praxis findet man Programme, die so schön linear ablaufen sehr selten. Vielfach müssen Entscheidungen getroffen werden, ob etwas so oder so gemacht werden soll. Auch Wiederholungen bestimmter Schrittfolgen sind an der Tages-Ordnung. Dazu aber später (→ [Programme mit Schleifen / Wiederholungen / Schlaufen](#)).

Betrachten wir zuerst einmal Verzweigungen im Programm-Ablauf. An bestimmten Stellen macht es oft keinen Sinn den "normalen" Ablauf weiter zu verfolgen. Nur wenn eine bestimmte Bedingung erfüllt ist, soll weiter gearbeitet werden. So ein Fall ist z.B. eine Division durch Null. Diese ist nicht definiert. Sollte so ein Fall in unserem Programm-Ablauf auftreten, dann ist jede folgende Berechnung (mit dem "Ergebnis der Division") unsinnig. Also sollte das Programm lieber anhalten, als Unsinn auszurechnen.

Häufig werden zwei verschiedene Strukturen unterschieden. Sachlich sind es beide Verzweigungen, aber bei der einfachen oder einseitigen Verzweigung wird nur der DANN- bzw. THEN-Zweig genutzt.

Bei vollständigen oder zweiseitigen Verzweigungen wird sowohl der DANN- oder THEN-Zweig, als auch der SONST- oder ELSE-Zweig in die Struktur aufgenommen. Obwohl beide Seiten / Fälle vorhanden sind, müssen sie nicht automatisch auch mit Programm-Code gefüllt werden.

Passender ist es natürlich, sonst würde man ja auch mit einer einseitigen Verzweigung hinarbeiten.

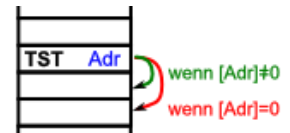
Als erste Variante schauen wir uns einseitige Verzweigungen an. Z.B. soll bei einem Fehler-Fall einfach nicht mehr weiter gearbeitet werden. Wir nehmen mal das Beispiel, dass eine Speicher-Zelle ein 0 enthält und dies sei nicht zulässig / definiert für unser Programm. Es soll dann stoppen.



Struktogramm-Symbol einer einfachen Verzweigung (bedingte Ausführung)

Test-Sequenz			
Speicher-Adresse	Assembler-Befehl (Asm)	Operand (Opnd)	Kommentare / Beschreibungen / Hinweise
00x			
00x +1	TST	031	Test, ob 2. Datum 0 ist → wenn ja, überspringen des nächsten Befehls
00x +	00	000	<i>nicht relavante Zell-Inhalte sind ausgegraut</i>
030	00	024	1. Datum
031	00	000	2. Datum
032	00	000	Ergebnis-Adresse
033	00	000	

Problem ist, für für den Test-Fall "[Adr] ≠ 0" nur einen folgenden Maschinen-Befehl nutzen können. Dahinter ist schon der automatische Weiter-Lauf für "[Adr] = 0". Hier wird quasi mit dem WAHR-Zweig fortgesetzt.



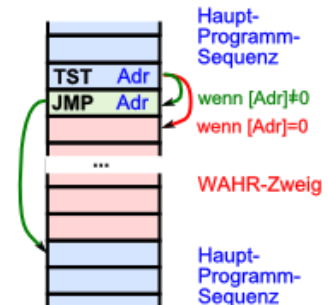
Somit bleibt uns nichts anderes übrig, als einen Sprung-Befehl zu benutzen.

Mit diesem überspringen wir den WAHR-Zweig (DANN oder THEN), um die Haupt-Programm-Sequenz (**bläulich**) weiter fortzusetzen.

Es folgt somit der WAHR-Zweig (**rötlich**). Dieser wird immer dann ausgeführt, wenn der Test der Inhalts der angegebenen Adresse Null ergeben hat.

Nach dem Durchlauf dieser Sequenz geht der Ablauf automatisch in die Haupt-Programm-Sequenz über.

Der alternative Zweig (FALSCH-Zweig) wird in unserem Fall nicht gebraucht. Er besteht nur noch aus dem Sprung-Befehl (**grünlich**) zum Überspringen des WAHR-Zweig's.



### Beispiele für Anwendungen einer bedingten Ausführung / einfachen Alternative:

nur fortsetzen, wenn eine Zahl größer als die andere ist

eine 1 in eine Zelle schreiben, wenn die Zahl größer als eine andere ist

das Maximum von zwei Zahlen in eine Zelle schreiben

ist  $a < b$  dann soll eine 1 in eine Zelle geschrieben werden, bei  $a = b$  eine 2 und wenn  $a > b$  dann eine 3

multiplikation einer Zahl mit einem festen Faktor (z.B.: 3)

testen ob 2 Zellen mit 0 belegt sind

Division teilbarer Zahlen

Modulo Berechnung (Rest bei ganzzahliger Division)

Ganzzahlige Division mit Rest (Ergebnis: Teiler und Rest)

Q: <https://www.inf-schule.de/rechner/johnny/spruenge/uebungen> (+ dre)

### Aufgaben:

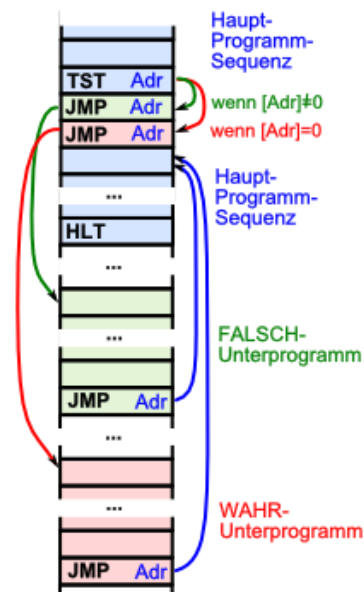
- 1.
2. Erstellen Sie ein Programm, das von den Daten aus der Speicherzelle 020 und 022 den größeren Inhalt in die Zelle 021 speichert!
3. Erstellen Sie ein Programm, das von den Daten aus der Speicherzelle 020 und 022 den kleinsten Inhalt in die Zelle 021 speichert!

Beim genauen Betrachten sind unsere Verzweigungen immer zweiseitig. Oft wird auch der zweite Zweig (FALSCH-, FALSE- od. SONST-Zweig) gebraucht.

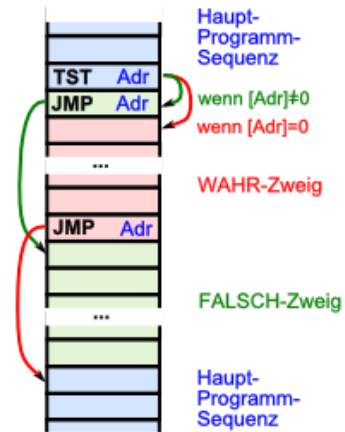
Das erfordert eine etwas kompliziertere Struktur unseres Programm's. Wie vorne besprochen folgt dem Jump-Befehl direkt hinter dem Test, der WAHR-Zweig (**rötlich**). Am Ende muss nun allerdings ein Sprung hinter den FALSCH-Zweig (**grünlich**) stehen. (Ansonsten würde auch der FALSCH-Zweig mit abgearbeitet werden!)

Der Sprung (direkt) hinter dem Test führt nun zum FALSCH-Zweig (**grünlich**). Nach dessen Durchlauf geht die Abarbeitung wieder automatisch in die Haupt-Programm-Sequenz (**bläulich**) über.

Liegen die WAHR- und FALSCH-Zweige irgendwoanders im Speicher – quasi als Unterprogramme – dann muss auch noch der Rücksprung in die Haupt-Programm-Sequenz bedacht werden.



Struktogramm-Symbol einer vollständigen Verzweigung



Wir suchen als Beispiel das Maximum oder die Größte zweier Zahlen (X, Y). In höheren Programmier-Sprachen gibt dafür entweder eine passende Funktion oder man löst die Aufgabe mit einem direkten Vergleich der beiden Zahlen. Leider bietet uns Johnny keinen passenden Befehl.

Man kann sich aber mit dem Bilden der Differenz (X - Y) zwischen beiden Zahl behelfen. Es kommt entweder eine Differenz größer als Null heraus. In dem Fall, war X die größere Zahl. Diese muss dann in eine Ergebnis-Zelle gespeichert werden.

Im anderen Fall – also die Differenz ist kleiner oder gleich Null – speichern wir Y in die Ergebniss-Zelle.

Da in Johnny die Differenz nie kleiner als Null werden kann, ergibt die Differenz-Bildung dann nur Null.

Bestimmen des Maximum's von zwei Zahlen			
Speicher-Adresse	Assembler-Befehl (Asm)	Operand (Opnd)	Kommentare / Beschreibungen / Hinweise
000	TAKE	020	lade 1. Zahl (aus Zelle 020) in den Akkumulator
001	SUB	021	subtrahiere vom Akkumulator den Wert aus Zelle 021
002	SAVE	021	speichere die Differenz (im Akku.) in die Zelle 021
003	TST	021	teste / vergleiche die Differenz (Zelle 021) mit Null
004	JMP	007	ist die Diff. > 0, dann springe zur Übernahme der 1. Zahl
005	TAKE	021	sonst (Diff <= 0), dann übernehme die 2. Zahl
006	JMP	008	beende Zweig und springe zum Programmende
007	TAKE	020	übernehme die 2. Zahl in Akkumulator
008	SAVE	022	nehme Akku als Maximum (Zelle 022)
009	HLT		Ende des Programm's
010			
	00	000	<i>nicht relavante Zell-Inhalte sind ausgegraut</i>
020	00	014	1. Zahl
021	00	033	2. Zahl
022	00	000	Ergebnis-Adresse (Maximum)
023	00	000	

### Übungs-Aufgaben:

1. Das Maximum-Programm soll kompakter werden und die Daten ab der Zelle 010 liegen. Passen Sie das Programm an und testen Sie es mit verschiedenen Zahlen!
2. Verändern Sie das Maximum-Programm so, dass es das Minimum der zwei Inhalte in der Zelle 021 speichert!
3. In den Speicherzellen 030 und 031 befinden sich zwei Zahlen. Ein Programm soll nun in die Zelle 029 eine 1 schreiben, wenn die Zahl in 031 größer als die in 030 ist. Ansonsten soll in der Zelle 030 eine 0 stehen.
4. Erstellen Sie ein Programm, dass eine Zahl in der Speicherzelle 025 mit 7 multipliziert! Das Ergebnis soll in der Zelle 026 auftauchen!
5. Eine teilbare Zahl in der Speicherzelle 030 soll durch die Zahl in Zelle 031 geteilt werden! Das Ergebnis speichern Sie in der Zelle 033!

### Übungs-Aufgaben für das gehobene Anspruchsniveau:

6. Realisiere die ganzzahlige Division mit Rest für zwei Zahlen in den Speicherzellen 040 (Dividend) und 041 (Divisor). Dem Quotient(enwert) ist die Speicherzelle 042 und dem Rest die Zelle 043 zugeordnet worden.
7. Erstellen Sie ein Programm, dass eine Zahl in der Speicherzelle 005 mit 17 multipliziert und in der Zelle 006 abspeichert!
8. Überlegen Sie sich ein Programm, dass eine Potentierung durchführen kann! Die Basis steht in der Zelle 050 und der Exponent in 051. Für das Ergebnis benutzen Sie 053!

## Programme mit bedingten Schleifen / Wiederholungen / Schlaufen

Wiederholungen sind das Lieblings-Geschäft der Programmierer. Sie unterstützen ihr allgegenwärtiges "Arbeits-Prinzip" Faulheit. Viele – wenn nicht gar die meisten Erfindungen (einschließlich Computer) und Software-Produkte sind deshalb erfunden worden, weil wir zu "faul" waren, es anders / traditionell / händisch zu machen (;-).

Genauso scheuen es die Programmierer gleichen Quelltext mehrfach hintereinander aufzuschreiben. Zu groß ist dabei die Gefahr, Abschreib-Fehler zu machen, oder bei Fehler-Korrekturen eine Abschrift zu übersehen.

Meist ist allerdings nicht klar, wieoft eine Schleife durchlaufen werden muss. Denken wir z.B. an die Multiplikation von zwei Zahlen.

Da Multiplikationen nicht im Befehls-Umfang von Johnny sind, müssen wir sie auf Additionen zurückführen. Das ist ein klassischer Fall für eine bedingte Schleife. Die Schleife (Addition der zu multiplizierenden Zahl) müssen wir sooft ausführen, wie es der zweite Faktor angibt. Irgendwann müssen wir testen, ob wir schon genug Additionen ausgeführt haben. Da unser Test-Befehl nur mit Null vergleichen kann, benutzen wir zur Kontrolle des zweiten Faktors nicht das Hochzählen, sondern das Runterzählen bis Null.

Multiplikation zweier Zahlen			
Speicher-Adresse	Assembler-Befehl (Asm)	Operand (Opnd)	Kommentare / Beschreibungen / Hinweise
000	NULL	020	Löschen der Ergebnis-Zelle
001	TAKE	020	lade den letzten Ergebnis-Stand in den Akkumulator
002	ADD	021	addiere den ersten Faktor zum Akku (letztes Erg.)
003	SAVE	020	speichere neues Erg. (im Akku) in die Ergebnis-Zelle
004	DEC	022	verringere den 2. Faktor um 1 (Runterzählen der Add.)
005	TST	022	prüfen, ob schon bei 0 angekommen
006	JMP	001	wenn nicht, dann springe zu Zelle 001 (wiederhole ab 001)
007	HLT	000	Programm beenden
008	00	000	<i>nicht relevante Zell-Inhalte sind ausgegraut</i>
020	00	000	Ergebnis
021	00	004	1. Faktor
022	00	005	2. Faktor

In den meisten Fällen wird man wohl zurück zu einer niederen Adresse springen und schon einmal getätigte Befehle wiederholen. Man kann aber auch ein Unterprogramm irgendwo im Speicher positionieren, zu dessen Anfang man dann springt- Am Ende des Unterprogramms muss dann ein Rücksprung erfolgen. Dieser wird sehr wahrscheinlich der gerade getätigte Test sein (muss es aber nicht!).

### Aufgaben:

1. *Schreiben Sie ein Programm, dass von einer Zahl in der Zelle 030 (z.B. = 24) sooft den Wert aus der Zelle 031 (z.B. = 4) abzieht, wie es in der Zelle 032 (z.B. = 3) angegeben wurde!*

- 2.
- 3.



## Übungs-Aufgaben:

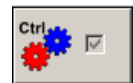
- 1.
- 2.
- 3.

### 1.2.2.2.2. Beobachtung des VON-NEUMANN-Befehls-Zyklus

Nachdem wir nun große Assembler-Programme geschrieben haben kehren wir noch einmal zu den Mikro-Befehlen und dem VON-NEUMANN-Befehls-Zyklus zurück.

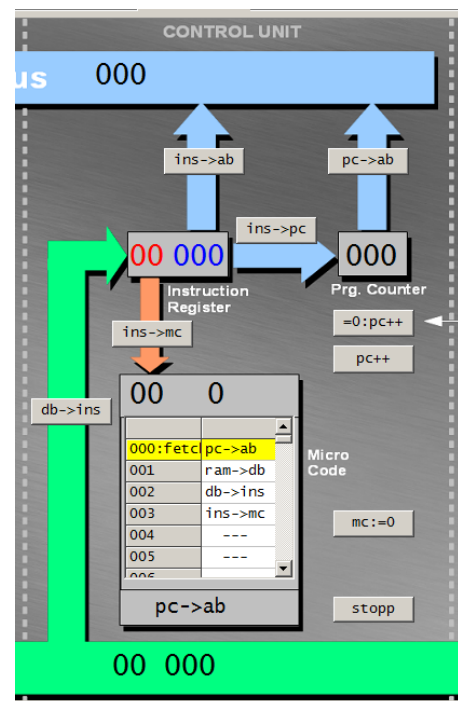
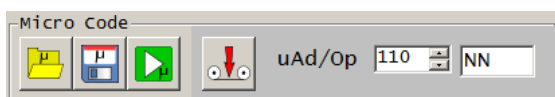
Der VON-NEUMANN-Befehls-Zyklus setzte sich ja aus den 5 Phasen FETCH, DECODE, FETCH OPERANDS, EXECUTE und WRITE BACK zusammen.

Für die Beobachtung der Abläufe decken wir die Control Unit über "Ctrl"-Schaltfläche wieder auf.



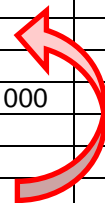
Als einfaches Beispiel betrachten wir ein gut bekanntes Assembler-Programm – die Addition von zwei Adress-Inhalten und dem Abspeichern des Ergebnis in einer weiteren Speicherzelle. Um einen kleinen Effekt zu zeigen "erweitern" wir das Programm vorm Halt noch um einen NOP-Befehl.

Der Simulator Johnny stellt uns nach dem Öffnen der Controll-Einheit einige zusätzlich Bedien-Elemente im Menü zur Verfügung. Der grüne Abspield-Button realisiert eine Abarbeitung eines Assembler-Befehls in Micro-Schritten. Das sind wieder genau die micro-Befehle, die wir zu Anfang (→ [Rechenwerk \(ALU ... Arithmetic Logic Unit \(Arithmetik-Logik-Einheit\)\)](#)) besprochen haben.



Die Micro-Befehle zu den einzelnen Assembler-Befehlen sind im Micro-Code des Prozessors gespeichert und werden dann Takt-gesteuert abgearbeitet.

Asm	Micro-Code	Instr.-Reg.	Prg.-Ctr.	ALU	MEM / RAM		
				Acc	010	011	012
fetch	pc -> ab	00 000	000		005	003	
	ram -> db						
	db -> ins	01 010					
	ins -> mc						
TAKE 010	acc:=0			000			
	ins -> ab						
	ram -> db						
	plus			005			
fetch	pc++		001				
	mc:=0						
	pc -> ab						
	ram -> db						
ADD 011	db -> ins	02 011					
	ins -> mc						
	ins -> ab						
	ram -> db						
fetch	plus			008			
	pc++		002				
	mc:=0						
	pc -> ab						
SAVE 012	ram -> db						
	db -> ins	04 012					
	ins -> mc						
	ins -> ab						
fetch	acc -> db						
	db-> ram						008
	pc++		003				
	pc -> ab						
NULL 000	ram -> db						
	db -> ins	09 000					
	ins -> mc						
	ins -> ab			000			
fetch	acc:=0						
	acc -> db						
	db -> ram						
	pc++		004				
fetch	mc:=0						
	pc -> ab						
	ram -> db						
	db -> ins	10 000					
HLT 000	ins -> mc						
	stopp						
	mc:=0						



**Aufgaben:**

1. Warum wiederholt die CPU die letzten Micro-Code's (roter Pfeil) immer wieder? Sollte die CPU nicht anhalten? Erläutern Sie Ihren Standpunkt!
2. Stimmt das Johnny-Assembler-Modell mit dem VON-NEUMANN-Zyklus-Modell überein? Prüfen Sie! Klären Sie ev. vorhandene Widersprüche auf!
- 3.

## VON-NEUMANN-Befehls-Zyklus

Schritt	Bezeichnung	Inhalt / Arbeits-Leistung
1	<b>(Instruction-) FETCH (IF)</b>	Befehls-Aufruf Befehl wird aus der Speicher-Adresse geladen, die in Befehls-Register (Befehls-Zähler) steht
2	<b>(Instruction-) DECODE (ID)</b>	Dekodierung Befehlszähler um die Befehls-Länge erhöht das Steuerwerk schaltet die zum Befehl gehörenden Logik-Schaltungen ein (bzw. gibt Daten-Leitung dahin frei)
3	<b>FETCH OPERANDS (FO)</b>	Operanden-Aufruf aus dem Speicher oder anderen Registern werden zusätzliche Operanden (Daten) geladen (auf die Logik-Schaltungen geleitet)
4	<b>EXECUTE (Execution Stage) (EX)</b>	Befehls-Ausführung Rechen-Logik arbeitet; Befehlszähler wird erhöht bzw. bei Schleifen / Sprüngen auf anderen Wert (neue Befehls-Adresse) gesetzt
5	<b>WRITE BACK (WB)</b>	Zurückschreiben (des Ergebnisses) ev. werden die Ergebnisse aus der Logik-Schaltung in Register oder Speicherzellen geschrieben

### 1.2.2.2.3. Erstellen neuer Makro's / Assembler-Befehle

z.B. CPY Quell-Adresse (Ziel-Adresse ist in der nachfolgenden Adresse (im Programm-Code))

Q: [https://www.inf-schule.de/rechner/johnny/zusatzmaterial/exkurs\\_eigene\\_makrobefehle](https://www.inf-schule.de/rechner/johnny/zusatzmaterial/exkurs_eigene_makrobefehle)

speichere Wert in einer bestimmten Adresse

---

**komplexe Aufgaben zu Johnny:**

- 1.
2. Was würde passieren, wenn ein Programm den ganzen Speicher mit Nullen beschreibt? Erklären Sie ausführlich! Macht es einen Unterschied, ob man von der untersten (000) zur obersten (999) löscht oder umgekehrt?
3. Erstellen Sie ein Programm, das ab einer bestimmten Speicherzelle die Werte aufsummiert!
  - a) für eine erste Variante in die Anzahl der Summanden festgelegt
  - b) in einer erweiterten Variante soll solange summiert werden bis, 0 in der zu lesenden Speicherzelle steht!
4. Schreiben Sie ein Programm, das beginnend bei einer vorgegebenen Speicherzelle solange die verdoppelten Speicher-Werte wieder abspeichert (auf der gleichen Speicherzelle), bis eine Null im Speicher gefunden wird!
- 5.
- 6.
7. Geben Sie in die Speicherzellen 040, 041 und 042 die Zahlen 24, 4 und 13 ein! Erstellen Sie ein Programm, das diese drei Zahlen addiert und das Ergebnis in Zelle 045 abspeichert! Speichern Sie sich das Programm als add3 ab! Notieren Sie sich untereinander die Befehle und machen Sie sich daneben Bemerkungen, was jeweils bei dem Schritt gemacht wird!
8. Erstellen Sie ein Programm dass die Inhalte der Speicherzellen 041 und 042 von der Zelle 040 subtrahiert! Das Ergebnis soll nun in der Speicherzelle 026 gespeichert werden! Speichern Sie sich das Programm als sub3 ab! Notieren Sie sich wieder die Befehle und passende Bemerkungen!
9. Übernehmen Sie die folgende Tabelle und berechnen Sie den Wert der Zelle 046 voraus, wenn jeweils das Programm sub3 abläuft!  
Belegen Sie die Speicherzellen 040, 041 und 042 immer jeweils neu mit den angegebenen Zahlen und lassen Sie jeweils das Programm laufen!

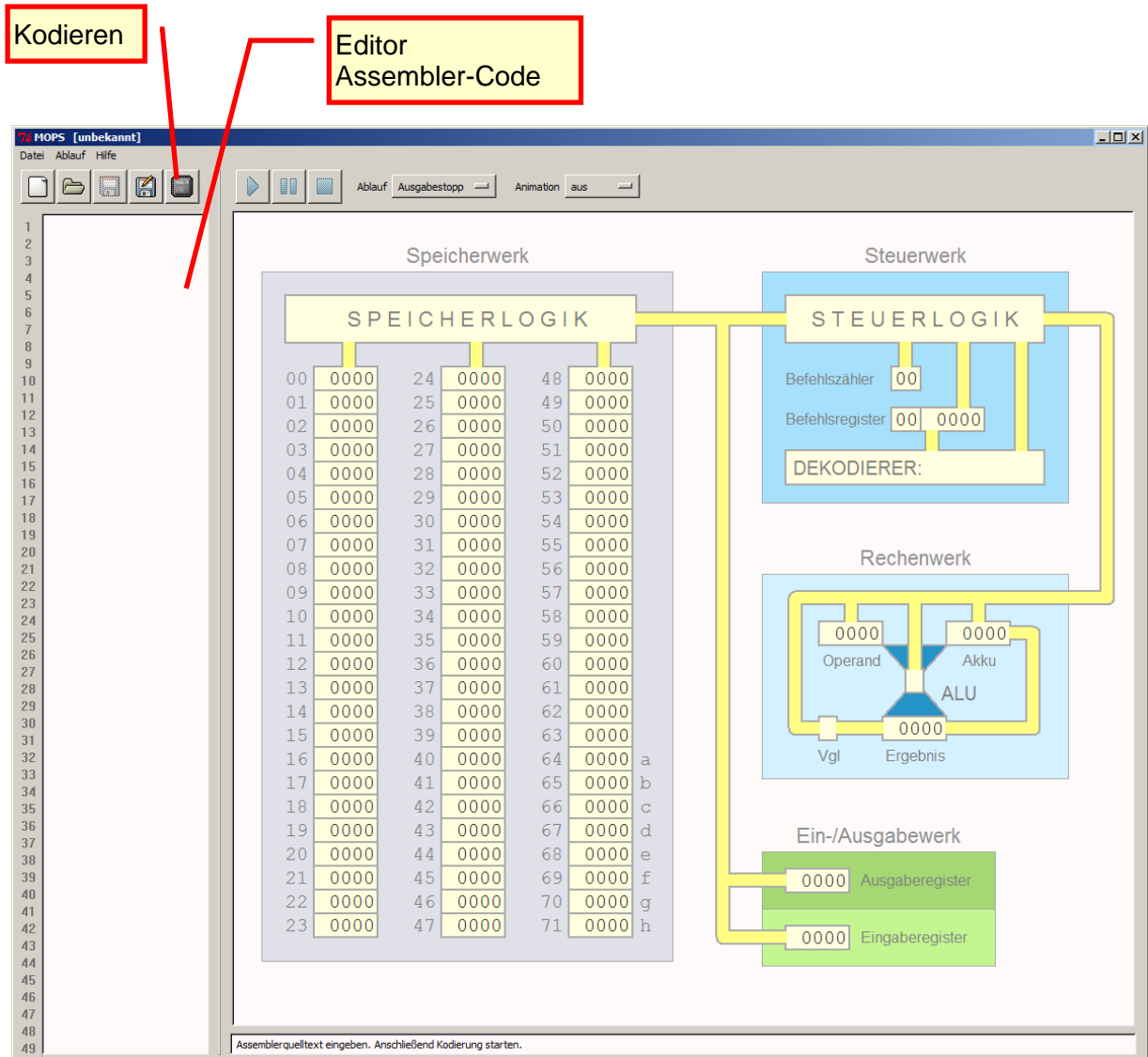
Speicher-Belegung			vorberechnetes Ergebnis	Speicherzelle	Vergleich / Bemerkungen
040	041	042		046	
30	17	4			
21	15	6			
23	20	7			

10. Bei der geringen Anzahl an Funktionen stellt sich die Frage, ob eine solche ALU prinzipiell alle Aufgaben lösen kann? Diskutieren Sie in der Gruppe!
- komplexe Aufgaben für die gehobene Anspruchsebene:**

11. Kombinieren Sie die Programme sub3 und add3 und lassen Sie dann in Zelle 044 die Addition der beiden Teilergebnisse (Summe und Differenz) eintragen!!
- x.
- y. Schreibe eine Virus für einen Johnny-Rechner, der sich selbst im Speicher kopiert!
- z.

### 1.2.2.3. Simulation eines VON-NEUMANN-Rechner mit MOPS

Bei dem Programm MOPS handelt es sich um einen **Modell-Rechner** mit **Pseudo-Assembler**. Der name leitet sich aus diesen beiden Haupt-Merkmalen ab. In MOPS wird eine VON-NEUMANN-Rechner (VNR) simuliert. Die dargestellten Abläufe orientieren sich an dem klassischen VON-NEUMANN-Zyklus.



Der Modell-Rechner hat einen begrenzten Speicher. Der erste Teil ist für den Programm-Code (Maschinen-Code in Dezimal-Darstellung) reserviert. Die Maschinen-Befehle sind ausgewählte – aber systematisch verwendete – Dezimal-Zahlen. Am Ende des Speichers sind für die Variablen a bis h insgesamt 8 Speicherzellen deklariert. Jede Speicherzelle kann dezimale Werte von -9999 bis +9999 aufnehmen.

Die Verwendung von Dezimal-Zahlen ermöglicht einen Einstieg in das Programm ohne die Schwelle Umwandlung / Verständnis von Dual- und / oder Hexadezimal-Zahlen.

Für Windows gibt es ein Installations-Programm. Das Programm darf in der veröffentlichten Form beliebig genutzt und weitergegeben werden (Freeware).

---

Die Linux-Version setzt jeweils einen aktuellen Python-Interpreter voraus.

## **Besonderheiten zum Aufbau der einzelnen Werke**

### ***Speicherwerk***

Wie schon erwähnt ist der Speicher in MOPS sehr begrenzt. Er besitzt nur 72 Speicherzellen. Die ersten 64 sind für Programm-Code (in Maschinen-Sprache) reserviert. Der restliche Speicher (8 Speicherzellen) sind für Daten verfügbar. Der Zugriff kann direkt über die Adresse oder symbolisch über die Variablen a bis h erfolgen.

Der Speicherinhalt jeder Zelle wird als Dezimal-Zahl angezeigt. Das erleichtert das Verständnis und verbessert die Übersicht.

### ***Arithmetik-und Logik-Einheit (ALU, Rechenwerk)***

ist in echten Computern im Microprozessor eingebaut

die ALU enthält Speicherzellen für die Rechen-Logik, diese werden allgemein Register genannt

ein Register ist besonders herausgestellt. das ist der Akkumulator (übers.: "Zusammenführer"). Hier hinein kommen die Rechen-Ergebnisse. Der Akkumulator (kurz Akku) kann aber auch eine Eingabe für die Rechen-Logik enthalten. Nach dem Abarbeiten des Rechen-Befehls ist die Eingabe aber durch die Ausgabe – also das Ergebnis – ersetzt.

### ***Steuerwerk***

gehört ebenfalls in den Microprozessor

vornehmliche Aufgabe ist das Befehls-Zählen, was man sich auch als Zeiger auf die gültige Programm-Speicherzelle vorstellen kann.

Der in der aktuellen Programm-Speicherzelle stehende Maschinen-Code (hier eine Dezimal-Zahl) wird in das Befehls-Register geladen und dann "dekodiert". Das bedeutet innerhalb des Microprozessor's werden die passenden Leitungen aktiv gesetzt und die Befehls-Abarbeitung (Micro-Code des Prozessor's) ausgelöst.

Da jeder Maschinen-Befehl in MOPS aus zwei Teilen (Speicherzellen) besteht, wird der Befehlszähler normalerweise immer um 2 erhöht. Bei Sprung-Befehlen kann auch eine bestimmte Zahl angegeben sein. Diese entspricht der Position des angesprungenen Befehls (Sprung-Adresse).

Das Steuerwerk kontrolliert auch die Bus-Leitungen. Das meint, ob die Leitungen freigeschaltet sind und gültige Daten (Speicher-Adressen, Daten) enthalten.

Die verschiedenen Bus-Systeme (Adress-, Daten- und Steuer-Bus) werden nicht unterschieden und durch die (dunkel-)gelben Stränge beschrieben. Die Detail hierzu werden in MOPS ignoriert.

### ***Ein- und Ausgabe-Werk***

Die Ausgabe erfolgt über ein extra Register – quasi auch wie eine Speicherzelle.

Eine Eingabe erfolgt online bei der Simulation des übersetzten Programm's. MOPS bleibt bei einer erwarteten Eingabe stehen und ließt nach einem [Enter] das Eingabe-Register aus.

Eine Eingabe wird nur akzeptiert, wenn diese gültig ist. In anderen Fällen muss man die Eingabe wiederholen.

---

## Erstellen eines Assembler-Programm's

mnemonischer Assembler-Code

der Assembler-Code ist nicht-case-sensitiv, das bedeutet man kann Groß- und Klein-Buchstaben beliebig verwenden

das Ende eines Programm's wird mit dem Befehl **end** festgelegt

alle Assembler-Befehle erwarten maximal einen Operanden  
zweiter Operand ist im Allgemeinen der Akkumulator

in jede Zeile darf und kann nur ein Assembler-Befehl geschrieben werden  
Befehl und Operand werden durch mindestens ein Leerzeichen (oder Tabulator) voneinander getrennt

Kommentare beginnen mit einem Semikolon (;), alles was dann rechts davon steht, wird beim Compilieren übergangen

Sprungziele können Zeilennummern (des Editor's) oder eigene Marken sein

eine Zeilennummer wird mit einer Raute (#) gekennzeichnet

eine Marke wird an der Defintions-Stelle mit einem Doppelpunkt (:) notiert

Marken müssen mit einem Buchstaben beginnen, es können dann weitere Buchstaben und Ziffern folgen

(Speicher-)Adresse beginnen mit einem Dollar-Zeichen (\$)

für Daten gibt es einen eingeschränkten Speicherbereich von \$64 bis \$71 mit somit 8 Daten-Speicherzellen

für diese Speicherzellen / Adressen können die Variablen **a** bis **h** als Alias-Symbole benutzt werden

## Übersicht zum Assembler-Code in MOPS (Cheat sheet)

Befehl		Maschinen-Code	Befehls-Funktion
<b>ld</b> adresse <b>ld</b> variable	load	10	lädt den Wert aus der angegebenen Speicher-Zelle in den Akkumulator
<b>ld</b> wert	load	11	lädt den Wert in den Akkumulator
<b>st</b> adresse <b>st</b> variable	store	12	speichert den Inhalt des Akkumulator's in die angegebene Speicher-Zelle
<b>in</b> adresse <b>in</b> variable	input	20	überträgt den Inhalt aus dem Eingabe-Register in die angegebene Speicher-Z.
<b>out</b> adresse <b>out</b> variable	output	22	überträgt den Inhalt des Akkumulator's in das Ausgabe-Register
<b>out</b> wert	output	23	schreibt den Wert ins Ausgabe-Register
<b>add</b> adresse <b>add</b> variable	add	30	addiert den Wert aus der angegebenen Speicherzelle zum Akkumulator
<b>add</b> wert	add	31	addiert der Wert zum Akkumulator (Summe → Akku)
<b>sub</b> adresse <b>sub</b> variable	subtract	32	subtrahiert den Wert aus der angegeb. Speicherzelle vom Akkumulator (Differenz im Akku)
<b>sub</b> wert	subtract	33	subtrahiert den Wert vom Akkumulator (Differenz im Akku)
<b>mul</b> adresse <b>mul</b> variable	multiply	34	multipliziert den Wert aus der angegeb. Speicherzelle mit dem Akku (→ Akku)
<b>mul</b> wert	multiply	35	multipliziert den Wert mit dem Akku (Produkt im Akku)
<b>div</b> adresse <b>div</b> variable	divide	36	dividiert den Akku-Wert durch den Wert aus der angegeb. Speicherzelle (Quotient im Akku)
<b>div</b> wert	divide	37	dividiert den Akku-Wert durch den Wert (Quotient im Akku)
<b>mod</b> adresse <b>mod</b> variable	modulo	38	berechnet den Rest der ganzzahligen Division des Akku's durch den Wert aus der angegeb. Speicherz. (Rest → Akku)
<b>mod</b> wert	modulo	39	berechnet den Rest der ganzzahligen Division des Akku's durch den Wert (Rest → Akku)
<b>cmp</b> adresse <b>cmp</b> variable	compare	40	vergleicht den Wert aus der angegeb. Speicherzelle mit dem Akkumulator ( → Akku)
<b>cmp</b> wert	compare	41	vergleicht den Wert mit dem Akkumulator ( → Akku)
<b>jmp</b> zeile <b>jmp</b> marke	jump	50	springt zur Zeile oder der Marke (setzt Befehl-Zähler auf diesen Wert)
<b>jlt</b> zeile <b>jlt</b> marke	jump if less then	52	springt zur Zeile oder der Marke, wenn (nach cmp) der Akku kleiner als Null ist (setzt Befehl-Zähler auf diesen Wert)
<b>jeq</b> zeile <b>jeq</b> marke	jump if equal	54	springt zur Zeile oder der Marke, wenn (nach cmp) der Akku gleich Null ist (setzt Befehl-Zähler auf diesen Wert)
<b>jgt</b> zeile <b>jgt</b> marke	jump greater than	56	springt zur Zeile oder der Marke, wenn (nach cmp) der Akku größer als Null ist (setzt Befehl-Zähler auf diesen Wert)
<b>end</b>		60	beendet das Programm (→ STOP)

**:marke** ... Definition einer Marke / An-Sprung-Stelle

**;kommentar** ... beliebiger Text als Kommentar



## einige ausgewählte MOPS-Programme (lose Sammlung):

1	; Gerade Zahlen rückwärts
2	in a
3	; gerade machen
4	ld a
5	mod 2
6	cmp 0
7	ld a
8	jeq weiter
9	sub 1
10	; rückwärts zählen
11	st b :weiter
12	out b
13	sub 2
14	cmp 0
15	jgt weiter
16	end
aus Bedienungs-Anleitung von MOPS	

Zahl negieren	
1	;zahl negieren
2	in a
3	ld a
4	sub a
5	sub a
6	st b
7	out b
8	end
... :	
1	;zahl negieren
2	in a
3	ld a
4	mul -1
5	st b
6	out b
7	end
Q: <a href="http://informatik.rostfrank.de/info/lex03/mops.html">http://informatik.rostfrank.de/info/lex03/mops.html</a>	

FIBONACCHI-Fkt.	
1	ld 0
2	st b ;n-2
3	ld 1
4	st c ;n-1
5	st h ;ergebnis
6	in a ;eingabe
7	ld h :fib
8	add b
9	add c
10	st h
11	ld c
12	st b
13	ld h
14	st c
15	out h
16	ld a
17	sub 1
18	st a
19	cmp 0
10	jgt fib
21	end
Q: <a href="http://informatik.rostfrank.de/info/lex03/mops.html">http://informatik.rostfrank.de/info/lex03/mops.html</a>	

Test auf gerade Zahl	
1	;test gerade zahl
2	;wenn gerade dann 1 in Ausgabe, sonst 0
3	in a
4	;testen
5	ld a
6	mod 2
7	cmp 0
8	jeq gerade
9	jgt ungerade
10	ld 1 :gerade
11	st b
12	out b
13	jmp ende
14	ld 0 : ungerade
15	st b
16	out b
17	end :ende
nach Q: <a href="http://informatik.rostfrank.de/info/lex03/mops.html">http://informatik.rostfrank.de/info/lex03/mops.html</a>	

Kubik-Funktion	
1	;3. Potenz (bis 21)
2	in a
3	;gerade machen
4	ld a
5	mod 2
6	cmp 0
7	ld a
8	jeq weiter
9	sub 1
10	;rückwärts zählen
11	st b :weiter
12	out b
13	sub 2
14	cmp 0
15	jgt weiter
16	end
nach Q: <a href="http://informatik.rostfrank.de/info/lex03/mops.html">http://informatik.rostfrank.de/info/lex03/mops.html</a>	

1	;sortieren zweier zahlen
2	;h=help
3	in a
4	in b
5	ld a
6	cmp b
7	jlt ende
8	;tausch
9	ld a
10	st h
11	ld b
12	st a
13	ld h
14	st b
15	end :ende
Q: <a href="http://informatik.rostfrank.de/info/lex03/mops.html">http://informatik.rostfrank.de/info/lex03/mops.html</a>	

bis 10 hochzählen	
1	in a
2	ld a
3	st a :naechste
4	cmp 10
5	jgt ende
6	add 1
7	jmp naechste
8	st a :ende
	end

1	
2	
3	
4	
5	
6	
7	
8	

Division mit Nachkommastellen	
1	;Werte eingeben
2	in a ;Zähler
3	in b ;Nenner
4	
5	;Sichergehen, dass b nicht 0 ist
6	ld b
7	cmp 0
8	jeq #3
9	
10	;Division (ganzzahlig)
11	ld a :loop ;Anfang Schleife
12	div b
13	st d
14	out d
15	
16	;Rest ermitteln
17	ld a
18	mod b
19	st a ;Rest
20	cmp 0 ;fertig?
21	jeq end
22	
23	;mit 10 multiplizieren
24	mul 10
25	st a
26	jmp loop
27	
28	;Beenden
29	end :end
Q: <a href="https://www.gommehd.net/forum/threads/mops.560644/">https://www.gommehd.net/forum/threads/mops.560644/</a>	

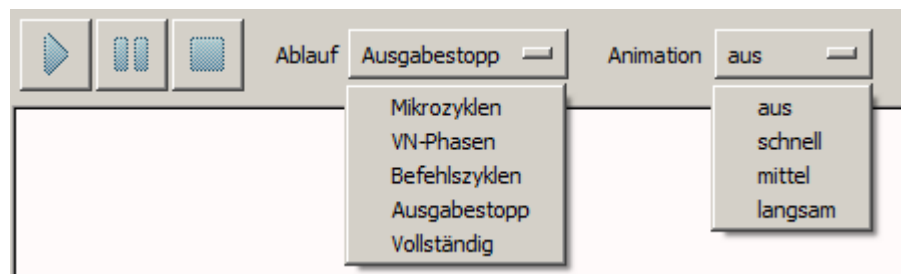
Division mit bel. vielen Nachkommast.	
1	;Werte eingeben
2	in a ;Zähler
3	in b :B ;Nenner
4	in c ;Anzahl der nachkommastellen
5	
6	;Sichergehen, dass b nicht 0 ist
7	ld b
8	cmp 0
9	jeq B
10	
11	;Division (ganzzahlig)
12	ld a
13	div b
14	st d
15	out d
16	
17	
18	;Rest ermitteln
19	ld a :loop ;Anfang Schleife
20	mod b
21	st a ;Rest
22	cmp 0 ;fertig?
23	jeq end
24	
25	;mit 10 multiplizieren
26	mul 10
27	st a
28	
29	ld a
30	div b
31	st d
32	out d
33	
34	ld h
35	add 1
36	st h
37	ld h
38	cmp c
39	jeq end
40	jmp loop ;Ende Schleife
41	
42	
43	;Beenden
44	end :end
Q: <a href="https://www.gommehd.net/forum/threads/mops.560644/">https://www.gommehd.net/forum/threads/mops.560644/</a>	

---

## Übersetzen des Programm's in Maschinen-Code und Simulation der Abläufe

"Kodieren" ([ F4 ]) übernimmt das Compilieren des Assembler-Code's in Maschinen-Code

klassische Steuerung über Steuer-Schaltflächen



VN-Phasen ... VON NEUMANN-Phasen (MOPS realisiert nur die Phasen 1 bis 3)  
zu beobachten in der Status.Zeile als Beschreibung und optisch in der Graphik

### **Aufgaben:**

**1. Erstellen Sie ein Assembler-Programm zur Addition von 2 Zahlen, die in a und b gespeichert werden sollen! Das Ergebnis soll in h stehen und ausgegeben werden.**

**2.**

**3.**

Q: <http://www.viktorianer.de/info/mops.html>

*Programme erkunden*

*Grenzen austesten*

*Umgang mit abweichenden Eingaben usw. (z.B. negative Zahlen) / Erkennen von Fehlern*

*Abändern eines Programm's (z.B. Ändern der Operation (Subtraktionen statt Addition))*

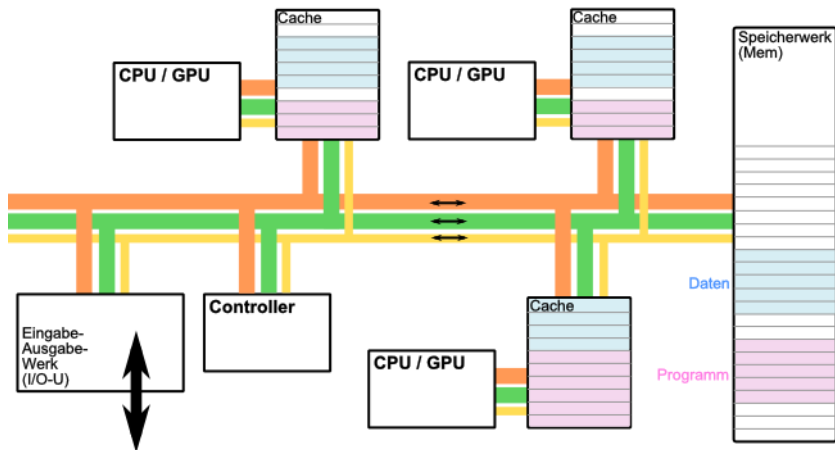
*Schreiben eigener Programme*

### 1.2.2.4. neuartige Strukturen / Modelle / Konzepte / Erweiterungen bei VON-NEUMANN-Rechnern

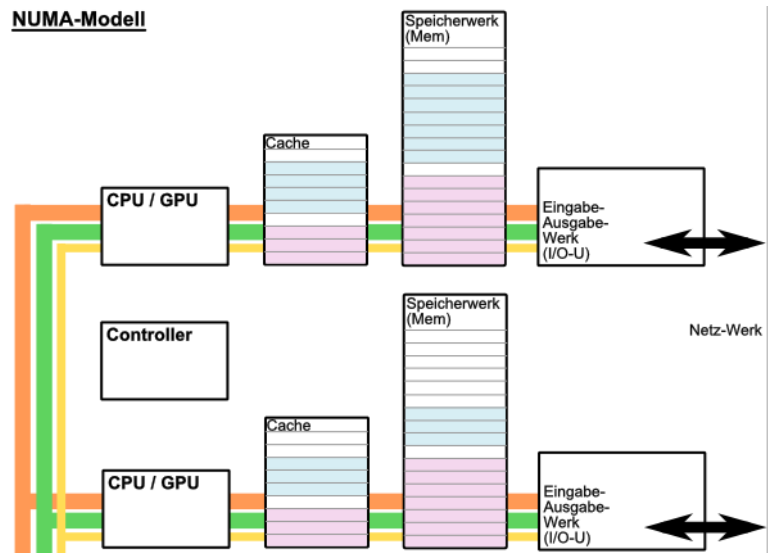
Basis sind VON-NEUMANN-Systeme, die durch Elemente anderer Rechner-Konzepte – meist Harvard-Konzept) ergänzt / erweitert werden  
 es werden die Vorteile der Einzel-Konzepte optimierend kombiniert

neu sind UMA- und NUMA-Computing-Modelle

**UMA-Modell**



**NUMA-Modell**



## 1.2.5. Harvard-Architektur

getrennte Programm- und Daten-Speicher mit getrennten Bus-System (natürlich trotzdem Adress-, Daten- und Steuer-Bus vorhanden, nur Datentyp ist unterschiedlich (Programme oder zu bearbeitende Daten))

historisch war der Programmspeicher ROM der Datenspeicher RAM

gleichzeitiges und unabhängiges Laden von Programmen (Befehlen) und Daten möglich  
System braucht weniger Arbeits-Takte

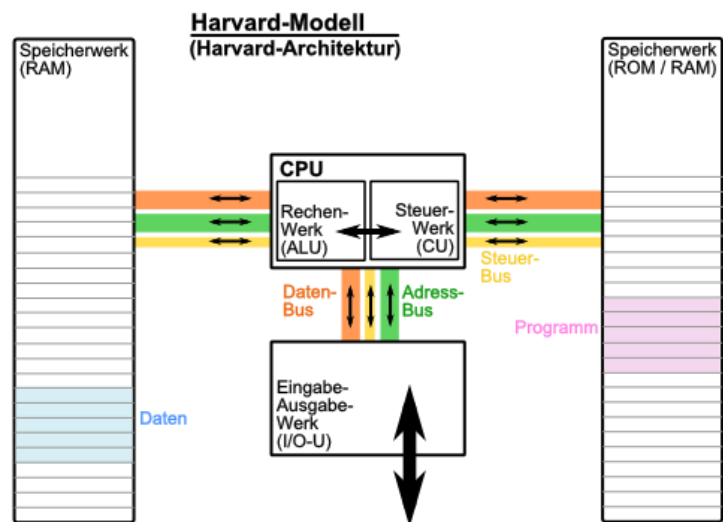
Speicherschutz ist weiterer Vorteil, Daten können nicht Programm-Code überschreiben (z.B. auch unempfindlicher gegen Viren)

keine Puffer-Überläufe möglich

nachteilig ist die notwendige großzügige Ausstattung mit beiden Speichern, weil sie sich nicht gegenseitig ergänzen können (wie das praktisch bei VN-Rechnern gemacht wird)

es kann zu nichtvorhersehbaren "Wettläufen" zwischen dem Programmteil und dem Datenteil kommen (Race Conditions)

heute werden wieder die Konzepte von der Trennung von Daten und Programmen in bestehende System integriert → Virtualisierung; Sandbox-Systeme



Theoretisch ist die Harvard-CPU der Flaschenhals dieses Modells. Da aber die CPU's technisch sehr hoch entwickelt sind und auch breiter (hinsichtlich der Daten- und Adress-Busbreite) ausgelegt werden können, gibt es keinen praktischen Flaschenhals.

### Definition(en): Harvard-Architektur

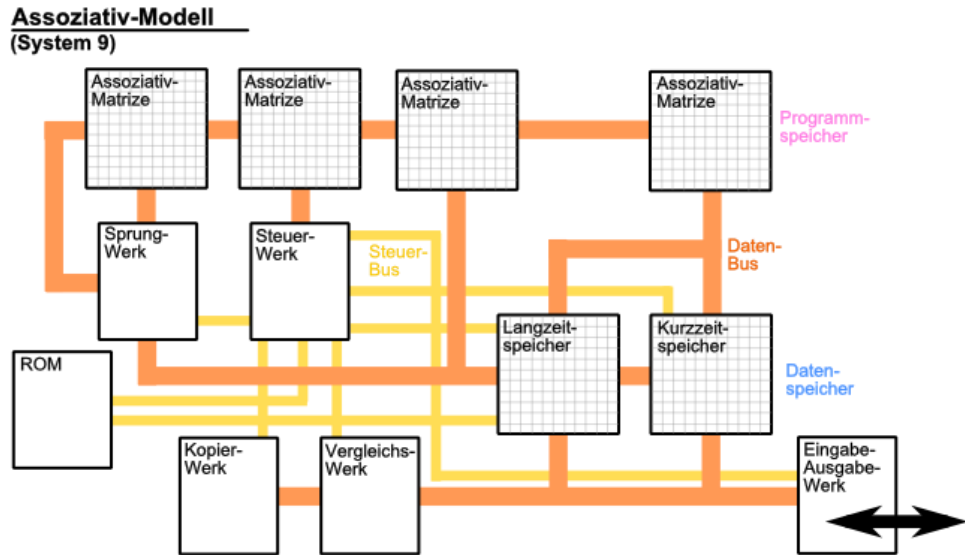
Die Harvard-Architektur ist ein Referenzmodell der Datenverarbeitung, bei der Programm- und Datenspeicher voneinander getrennt sind. Die (Harvard-)CPU hat zentralen Zugriff auf die beiden Speicher und die Input-Output-Werke.

Ein Harvard-Rechner ist ein Schaltungs- / Geräte-Konzept bei dem seine drei Grundgeräte (Programm- und Daten-Speicherwerk und Ein-Ausgabe-Werk) über einen zentralen Prozessor verbunden sind, gesteuert werden und miteinander kommunizieren.

## 1.2.6. Assoziativ-Maschine

Assoziativ-Matrizen sind der Speicher für Programme und Daten; keine festen Speicherort für Daten; arbeitet mit unterschiedlichen / wandelbaren / zufällig gewählten Strategien, so dass die Ergebnisse von Aufruf zu Aufruf / Programmabarbeitung zu Programmabarbeitung unterschiedlich ausfallen

angelehnt an Informations-Verarbeitung in biologischen System (Gehirn) allerdings ausschließlich mit Dual-System (im biologischen Systemen gibt es auch analoge Komponenten) praktisch immun gegen (heutige) Computerviren, Ausspähen von Daten extrem erschwert



### Definition(en): Assoziativ-Architektur

Die Assoziativ-Architektur ist ein offenes Modell der Datenverarbeitung, bei dem über eine freie Programmierung – assoziativ über nachfolgende Befehle hinweg – eine Verknüpfung der Daten realisiert wird, die in mehreren – unterschiedlich miteinander verbundenen – Matrixen-Speicherwerken gehalten werden.

---

## Aufgaben:

- 1.
2. **Geben Sie an bis zu welcher Grösse (hexadezimal) die 4, 8, 16, 32 und 64 bit-Systeme ihren Speicher adressieren können! Ermitteln Sie die dezimalen Äquivalente!**
3. **Ein befreundeter Nutzer hat mit seinem 32 bit-Windows-System (2 GB Hauptspeicher, 1 TB Festplatte, 1 TB Graphikkarte) bei speziellen Anwendungen Speicher-Probleme. Er möchte jetzt von Ihnen eine Empfehlung, wie er seinen Rechner + Anwendung sinnvoll (mit geringen Kosten) aufrüsten / verändern kann.  
Machen Sie ein oder zwei Vorschläge und erläutern Sie diese!**

## 1.2.7. Ternär-Rechner

### 1.2.7.1. Geschichtliches / Historie

1956 von Nikolai petrowitsch BRUSENZOW (1925 - ) als Rechner "Setun" realisiert arbeitete mit 18 Ternär-Ziffern (18 Trits (ternary digits)) namensgebend war der Fluß nahe der LOMONOSOW-Universität in Moskau

Herbert R. GROSCH entwickelte am MIT zur gleichen Zeit ein vergleichbares Projekt ("Whirlwind")  
Projekt hatte aber kein Erfolg

funktionierte mit magnetischen Schalt-Elementen und Dioden (Röhren wurden ausgeschlossen; Transistoren gab es noch nicht)

Bau-Elemente per Hand gefertigt und gelötet  
nach 10 Tagen erster Prototyp funktionsfähig und er funktionierte fehlerfrei

Speicher wurde als 6 Trit-System (= 1 Tryte)

50 Rechner produziert  
davon 30 an sowjetischen Universitäten genutzt

Projekt in der Sowjetunion weiterverfolgt und dort als Setun-70 realisiert (1970)

6'000 Operation pro Sekunde

1974 wurde noch ein Lern-System ("Mentor") entwickelt



Setun-Rechner  
Q: <https://alchetron.com/Setun>

Rechner Setun 70  
Q:

danach endete die Ära der Ternär-Computer

**Simulatoren:**

<https://github.com/askfind/Emulator-Setun-1958>

**1.2.7.2. Grundlagen / ternäre Logik**

es gilt:

- +1 ...WAHR
- 1 FALSCH
- 0 UNBESTIMMT

		NOT	
a	-1	+1	-1
	0	0	0
	+1	-1	+1

logische Operationen mit zwei Operanten

Junctor	Umschreibung(en)	Gatter	
<b>Tautologie</b>	immer WAHR		
<b>Exklusion</b>	SCHEFFERScher Strich	NUND, NAND	
<b>Implikation</b>			
<b>Replikation</b>			
<b>Disjunktion</b>	oder	ODER; OR	
<b>Pränonpendenz</b>	niemals a		
<b>Postnonpendenz</b>	niemals b		
<b>Äquivalenz</b>	exklusives Nicht-Oder	XNOR	Spiegelungs-
<b>Kontravalenz</b>	exklusives Oder	XOR	Achse
<b>Postpendenz</b>	nur b		
<b>Präpendenz</b>	nur a		
<b>Rejektion</b>	Nicht-Oder	NOR	
<b>Präsektion</b>	negative Replikation		
<b>Postsektion</b>	negative Implikation		
<b>Konjunktion</b>	Und	UND, AND	
<b>Antilogie</b>	immer FALSCH		

für einzelne Junctoren gibt es leicht andere Ergebnis-Tabellen

man muss sich aber für ein Logik-System (KLEENE K3 od. ŁUKASIEWICZ Ł3) entscheiden, dann ist sich in sich konsistent

hier Umsetzung der STILLER-Logik S3

		b		
AND		-1	0	+1
a	-1	-1	-1	-1
	0	-1	0	0
	+1	-1	0	+1

		b		
OR		-1	0	+1
a	-1	-1	0	+1
	0	0	0	+1
	+1	+1	+1	+1



		b		
XOR		-1	0	+1
a	-1	-1	0	+1
	0	0	0	0
	+1	+1	0	-1

		b		
NAND		-1	0	+1
a	-1	+1	+1	+1
	0	+1	0	0
	+1	+1	0	-1

		b		
NOR		-1	0	+1
a	-1	+1	0	-1
	0	0	0	-1
	+1	-1	-1	-1

		b		
XNOR		-1	0	+1
a	-1	+1	0	-1
	0	0	0	0
	+1	-1	0	+1

immer wenn, dann

		b		
(mat.) Implikation		-1	0	+1
a	-1	+1	+1	+1
	0	0	0	+1
	+1	-1	0	+1

nur wenn, dann

		b		
(mat.) Replikation		-1	0	+1
a	-1	+1	0	-1
	0	+1	0	0
	+1	+1	+1	+1

genau dann, wenn

		b		
		-1	0	+1
(mat.) Äquivalenz	-1	+1	0	-1
	0	0	0	0
	+1	-1	0	+1

		b		
		-1	0	+1
Postsektion	-1	-1	-1	-1
	0	0	0	-1
	+1	+1	0	-1

		b		
		-1	0	+1
Präsektion	-1	-1	0	+1
	0	-1	0	0
	+1	-1	-1	-1

		b		
		-1	0	+1
Tautologie	-1	+1	+1	+1
	0	+1	+1	+1
	+1	+1	+1	+1

		b		
		-1	0	+1
Antilogie	-1	-1	-1	-1
	0	-1	-1	-1
	+1	-1	-1	-1

		b		
		-1	0	+1
Präpendenz	-1	-1	-1	-1
	0	0	0	0
	+1	+1	+1	+1

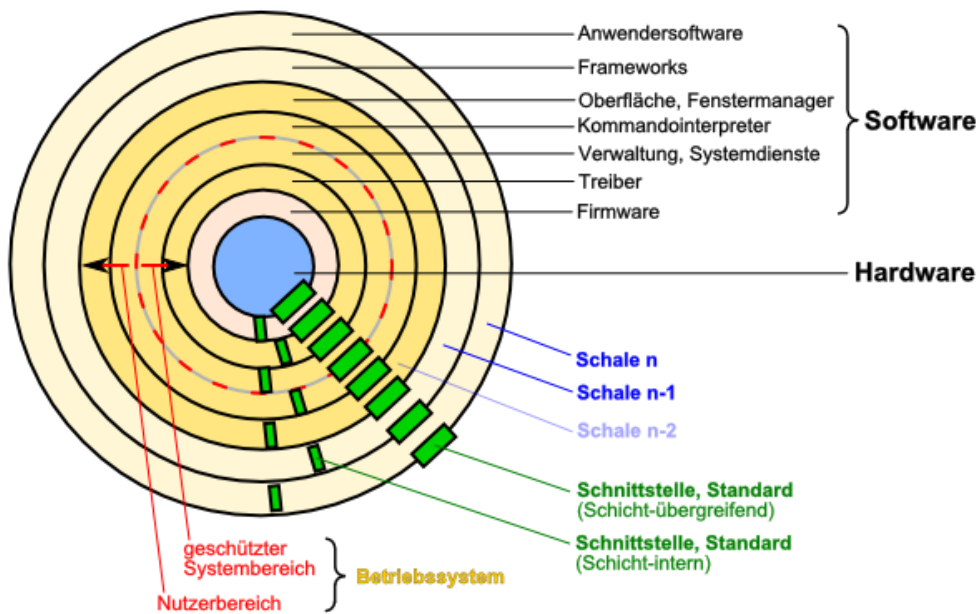
		b		
		-1	0	+1
Pränonpendenz		-1	0	+1
a	-1	+1	+1	+1
	0	0	0	0
	+1	-1	-1	-1

		b		
		-1	0	+1
Postpendenz		-1	0	+1
a	-1	-1	0	+1
	0	-1	0	+1
	+1	-1	0	+1

		b		
		-1	0	+1
Postnonpendenz		-1	0	+1
a	-1	+1	0	-1
	0	+1	0	-1
	+1	+1	0	-1

nach Q: STILLER, Joachim: Dreiwertige Logik – Zur dreiwertigen Logik

## 1.3. das Schalen-Modell



Schnittstellen sind definierte Objekte, Methoden (Funktionen) und Attribute (Variablen (für Kennwerte)) und deren Beschreibung.

So stellt Windows (als Betriebssystem) z.B. die verschiedenen Bedienelemente zur Verfügung. Deren Aussehen und Funktionalität ist in allen Windows-Programmen gleich. Ein Programmierer muss in seinem Programm nur festlegen, wo sich das Bedienelement im Programm-Fenster befindet und welche Eingangs- und Ausgangs-Daten sowie Optionen genutzt werden sollen.

Wie diese Bedienelemente funktionieren und welcher Quellcode dahintersteckt bleibt dem Programmierer verborgen. Sie sind das Betriebsgeheimnis von Microsoft.

Intern greifen die Bedienelemente auf rudimentäre Funktionen des Betriebssystems zurück. Auch diese sind definiert, damit andere Programmierer (von Microsoft) diese benutzen können. Solche Schicht-internen Schnittstellen sind ebenfalls ein Betriebsgeheimnis.

Die rudimentären Funktionen greifen wieder auf noch elementarere Funktionen der Treiber zurück. Die Treiber besitzen wiederum eine Schnittstelle zu den Hardware-orientierten Funktionen innerhalb dieser Schicht. Wie die Hardware genau programmiert wird ist das Betriebsgeheimnis der Herstellerfirmen.

### **Schnittstellen im Alltag:**

**im Auto: Lenker, Pedale und Schalter** (zum Bedienen des Autos)

wie das Auto intern (z.B. der Motor oder der Board-Computer) funktioniert, ist für den Fahrer nicht wichtig

**im Haushalt: Fernsteuerung zum Fernseher**

wir bedienen die Knöpfe, wie die Steuerbefehle zum Fernseher übertragen werden und wie der Fernseher intern z.B. die Kanäle oder die Lautstärke einstellt sind (für den Bediener) verborgene Funktionen der Fernseh-Elektronik

---

### im Smartphone: Browser-App

z.B. werden mittels Touch-Befehlen die Verbindung zu einer Webseite aufgebaut und die Steuerelemente der Webseite benutzt; wie der Verbindungs-Aufbau und die Daten-Übertragung funktioniert, weiss nur der Browser-Programmierer.

#### Aufgaben:

- 1. Finden Sie weitere Alltags-Schnittstellen! Beschreiben Sie kurz die Kommunikation zwischen den Schichten!*
- 2. Warum sind Schnittstellen-nutzende Programme eigentlich immer kleiner als solche Programme, die direkt auf die Hardware zugreifen!*

Schnittstellen sind genauso wie die Trennung von Inhalt und Design Beispiele für das sehr erfolgreiche Konzept **Separation of Concerns** (Trennung nach Belangen / Bedarf)

Beim Trennen von Inhalt und Design (/ Gestaltung) – wie es z.B. bei der Erstellung von Webseiten auftritt – liegt vorrangig eine horizontale Nebenordnung vor. Eine CSS-Datei (CSS ... Cascading Style Sheets) beschreibt das Aussehen der Inhalte und Gliederungen aus der HTML-Datei.

Bei den Schnittstellen liegt dagegen vorrangig eine vertikale Struktur der Auftrennungen nach den Belangen vor. Eine Schale (der Verantwortlichkeit und Kompetenz) legt sich um eine untergeordnete – elementarer und Hardware-näher angelegte Schale. Nach Außen folgt eine mehr Anwendungs- und Menschen-orientierte Schale, die auch komplexer strukturiert ist.

Schalen-Ebene	Bezeichnung	Beispiele	
oberste Ebene	höhere Programmiersprache	print-Befehl input-Befehl Schleifen-Anweisung ...	print("ABC")
			erzeugeBitMusterAufConsole("A") ...
	Betriebssystem		call Maschinen-Routine
	Hardware		setzt Bit's im Video-Speicher

---

## 1.4. Quanten-Computing

Q: basierend auf Open.HPI-Kursen "Einführung in Quantencomputing"; Prof. JUST; 2022 sowie "Quantenkryptographie"; Prof. HETTEL, 2022 sowie ...

### 1.4.0. Grundlagen

heute praktisch noch weitestgehend Zukunftsmusik  
Theorie schon recht entwickelt  
es fehlt die praktische Realisierung

#### 1.4.0.1. Historie / Geschichte

mit der Wende vom 19. zum 20. Jhd. kam die Quanten-Physik auf  
Erkenntnis, dass sich Elementar-Teilchen und Licht-Teilchen / -Quanten anders verhalten als  
die klassischen Objekte in der mechanischen Physik

Welle-Teilchen-Dualismus

mathematische Modelle der Quanten-Physik / Quanten-Mechanik schwer zugänglich  
PLANCK (→ Wirkungs-Quantum), EINSTEIN (→ "spukhafte Fernwirkung"), BOHR (→ Atom-  
Modell), HEISENBERG (→ Unschärfe-Relation), SCHRÖDINGER (→ Orbital-Gleichung; "Katze"),  
STERN, PAULI, BORN, FEYMAN

Berechnungs-Modell der Quanten-Informatik (1980 – 1985)

seit 1994 SHOR beschreibt Algorithmus zur Faktorisierung von Zahlen  
hätte große Auswirkung auf Kryptographie → großes öffentliches Interesse  
praktische Umsetzung noch offen, weil Hardware fehlt

Was könnten Quanten-Computer (besser)?

- völlig neuartige Dinge berechnen (, die auf klassischen Computern nicht berechenbar sind)
  - echter Zufall
  - Teleportation
  - abhörsichere Kommunikation
- Dinge berechnen, die auf klassischen Computern zwar berechenbar sind, aber viel zu lang dauern würden
  - schnelles Knacken von klassischen kryptographischen Verfahren (quasi: paralleles Brute force)
  - Optimierungs-Probleme
  - Material-Forschung
  - Klima-Modelle
  - Arzneimittel-Wirkungen
-

derzeit nur serielle Abarbeitung von Quanten-Verarbeitung  
praktisch wird aber eine parallele Verarbeitung gebraucht

großes Zukunfts-Potential

### 1.4.1. Qubit's

kleinste Information-Einheit eines Quanten-Computer's  
entspricht dem Bit in einem klassischen Computer

Schreibweise nach DIRAC oder auch Bra-Ket-Schreibweise (Wortspiel zu bracket = engl. Klammer) mit snekrechten Strich beginnend und mit schließender Winkel-Klammer (ersatzweise: Größer-Zeichen) endend:  $|0\rangle$

	(klassisches) Bit	Qubit
	<ul style="list-style-type: none"> <li>Wert 0 oder 1</li> <li>(aktueller) Zustand: 0 oder 1</li> </ul>	<ul style="list-style-type: none"> <li>Wert <math> 0\rangle</math> oder <math> 1\rangle</math> oder einen Wert dazwischen (Superposition)</li> <li>Zustand ist <math>\alpha 0\rangle + \beta 1\rangle</math></li> </ul>
	<ul style="list-style-type: none"> <li>z.B.: - ...0;   ...1</li> </ul>	<ul style="list-style-type: none"> <li>z.B.: - ...<math> 0\rangle</math>;   ...<math> 1\rangle</math>; aber auch: / ... irgendetwas dazwischen <math>\alpha 0\rangle + \beta 1\rangle</math></li> </ul>
<b>Eigenschaften</b>	<ul style="list-style-type: none"> <li>beim Auslesen (/ Messen) des Zustand's ändert sich der Wert nicht (Realismus)</li> <li>die Änderung eines Bit's kann nicht gleichzeitig ein anderes Bit beeinflussen (es wird immer Zeit / ein Takt gebraucht) (Lokalität)</li> </ul>	<ul style="list-style-type: none"> <li>das Auslesen (/ Messen) des Zustand's ändert meistens auch den Wert / Zustand <math>\rightarrow</math> ergibt immer <math> 0\rangle</math> oder <math> 1\rangle</math></li> <li>die Änderung eines Qubit's kann den Zustand unmittelbar / im selben Augenblick verändern (Verschränkung, "spukhafte Fernwirkung" EINSTEIN)</li> </ul>
<b>Algorithmen</b>	<ul style="list-style-type: none"> <li>verändert Bit für Bit das System</li> <li>laufen Schritt-artig / getaktet ab</li> <li>sind (per Definition) deterministisch (liefern bei gleichen Ausgangs-Bedingungen immer das gleiche Ergebnis)</li> </ul>	<ul style="list-style-type: none"> <li>eine Veränderung (ein Algorithmenschritt) eines Qubit's verändert i.A. unmittelbar das gesamte System</li> <li>probabilistisch (Zufalls-geprägt)</li> </ul>

$$|q\rangle = \alpha|0\rangle + \beta|1\rangle$$

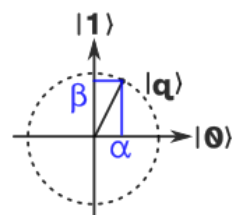
$\alpha$  und  $\beta$  sind komplexe Zahlen (also: Bereich / Menge  $\mathbb{C}$ )

als Nebenbedingung gilt:  $|\alpha|^2 + |\beta|^2 = 1$

Beschreibung als Matrix:

durch Basis-Vektoren:  $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$   $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

allgemein:  $|q\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$



Man kann sich ein Qubit auch als schwingendes Licht-Teilchen vorstellen. So ein Photon kommt der praktischen Umsetzung von Quanten-Computern mit Licht auch sehr Nahe. Das Photon ist dann ein schwingendes Teilchen. Liegt die Schwingungs-Ebene in der Waage-

rechten, dann ist dies im Basis-Zustand  $|0\rangle$  und schwingt es in der Senkrechten, dann entspricht dies dem  $|1\rangle$ -Zustand.

In den meisten Fällen haben wir eine Schwingungs-Ebene irgendwo zwischen Waagerechten und Senkrechten, was eben durch unsere Gleichung  $|q\rangle = \alpha|0\rangle + \beta|1\rangle$  ausgedrückt wird. Mit der Messung / Beobachtung nimmt ein Photon dann eine der beiden Basis-Zustände ein.

### Definition(en): Qubit (Quanten-Bit)

Ein Qubit ist ein Zweizustands-System mit Quanten-Eigenschaften, dass durch Messung sicher einen der beiden Zustände (hier:  $|0\rangle$  od.  $|1\rangle$ ) hat.

Ein Qubit ist die kleinste Rechen- / Informations-Einheit eines Quanten-Rechner's. Für das Qubit gelten die Gesetze der Quanten-mechanik, d.h. z.B., dass das Qubit bis zu seiner Messung mehrere Zustände einnehmen kann, die aber mit einer Messung eindeutig festgelegt sind.

im Qubit liegen die Zustände 0 und 1 quasi überlagert vor, erst wenn man misst, dann offenbart sich der aktuelle Wert  
wie bei SCHRÖDINGERS Katze, erst wenn man in die Box schaut weiss man, ob die Katze lebt oder vergiftet wurde

die Nachbildung von 45 Qubit's würden den Speicherplatz des heute größten Super-Rechners benötigen

bei 50 Qubit's gelangen wir an die Grenze der sogenannten Quanten-Überlegenheit die Rechen-Leistung eines 50 Qubit-Rechner's könnte mit keinem denkbaren Super-Rechner mehr erreicht werden

ein Rechner der 250 Qubit's nachbilden wollte, benötigt alle Atome des Universum's als Bit-Speicher

Qubit's	Nachbildung mit ... bit	
1	256	
2	512	
3	1024	
4	2048	
5	4096	
6	8192	
7	16384	
8	32768	
10	131072	
20	$1,71799 * 10^{10}$	
30	$2,25180 * 10^{15}$	
50	$3,86856 * 10^{25}$	
100		

### Modell für einen Quanten-Algorithmus:

Würfel mit Röhren als Kanten und Hohlkugeln als Ecken  
eine Kugel ist mit Flüssigkeit gefüllt

Würfel kann in x-, y- und z-Richtung geschüttelt werden, um die Flüssigkeit zu verteilen  
ein Arbeitsschritt ist einmal Schütteln, wobei sich die Flüssigkeit über die Kante (der Schüttel-Richtung) gleichmäßig verteilt

Aufgabe: Flüssigkeit der einen Kugel gleichmäßig auf alle Kugel verteilen

? Wieviel Schüttel-Operationen sind notwendig, um die Flüssigkeit gleichmäßig auf die Kugeln zu verteilen?

Qubit-Eigenschaften



---

**Definition(en): Superposition**

Die Superposition ist die Fähigkeit eines (Quanten-)Objekt's, in mehreren Zuständen gleichzeitig zu sein.

Erst mit der Beobachtung / Messung nehmen die (Quanten-)Objekte einen definierten Zustand ein.

**Definition(en): Verschränkung**

Die Verschränkung ist die Eigenschaft von zwei kombinierten / verbundenen (Quanten-)Objekten, bei der Änderung des einen Objekt's instantan / praktisch gleichzeitig das zweite Objekt ebenfalls zu verändern.

**Definition(en): Dekohärenz**

Die Dekohärenz ist die Eigenschaft eines (Quanten-)Objekt's, bei der Wechselwirkung mit der Umgebung einen definierten Zustand einzunehmen und damit auch ihre Superposition-Eigenschaft (unwiederbringbar) zu verlieren.

wichtig für diverse Verrechnungen ist der Fakt, ob Qubit's **unterscheidbar** sind.

Für die Basis-Zustände  $|0\rangle$  und  $|1\rangle$  ist das eindeutig möglich. Was passiert aber mit Qubit's im allgemeinen / diversen Zustand  $\alpha|0\rangle + \beta|1\rangle$ ? Hier bewirkt erst die Messung einen definierten Zustand. Es kann der Zustand  $|0\rangle$  oder  $|1\rangle$  herauskommen. Misst man also z.B. eine  $|0\rangle$ , dann ist nicht klar, ob der Zustand vorher schon eine  $|0\rangle$  war oder erst durch die Messung aus dem diversen Qubit entstanden ist. Dieses Problem lässt sich auch mit Quanten-Gattern ( $\rightarrow$ ) nicht lösen. Qubit's sind nur bis zu einem bestimmten Teil unterscheidbar. Allgemein gilt das nicht.

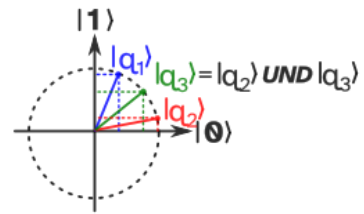
**Definition(en): Unterscheidbarkeit**

Die Unterscheidbarkeit beschreibt .

### 1.4.2. Quanten-Schaltkreise – Schaltkreise für Qubit's

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \Psi \dots \text{sprich: psi}$$

bei Quanten-Schaltkreisen entspricht die Anzahl der Eingänge (z.B.  $\alpha|0\rangle$ ,  $\beta|1\rangle$ ,  $|\Psi\rangle$ ) der Anzahl der Ausgänge



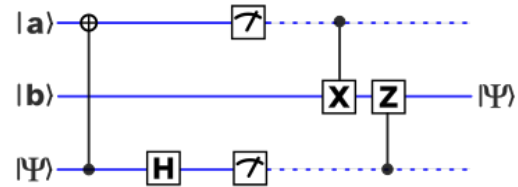
in einen Quanten-Schaltkreis (QSK) gehen genausoviele Eingänge hinein, wie hinauskommen

Beispiel für einen Quanten-Schaltkreis:

Er besteht aus den Quanten-Gattern CNOT, H, X und Z. Dazu gibt es Mess-Punkte, die entweder mit einem Analog-Meßgeräte-Logo oder M gekennzeichnet werden.

Nach der der Messung ist der Zustand entweder  $\alpha|0\rangle$  oder  $|1\rangle$ . Dies wird durch die gestrichelten Linien angegeben.

Nach dem Messen verhalten sich die Qubit's wie normale Bit's.



### Berechnungs-Modell nach FEYMAN und DEUTSCH

genutzte Quanten-Eigenschaft ist die Verschränkung  
 derzeit lassen sich rund 20 bis 130 Qubit's modellieren  
 einzelne Qubit's lassen sich beeinflussen  
 Lösung universeller Probleme denkbar

### weiteres Modell: Quantum Simulated Annealing (z.B. D\_Wave)

modelliert bis zu 5'000 Qubit's  
 keine Kontrolle über einzelne Qubit's  
 genutzter Quanten-Effekt ist der Quanten-Tunnel-Effekt  
 derzeit für die Lösung von Optimierungs-Problemen genutzt  
 Prinzip:  
 ein mit dem "Problem" vorbereitetes System wird bis dicht zum absoluten Null-Punkt abgekühlt  
 das System nimmt nun den geringst-möglichen Energie-Zustand (für das "Problem") ein

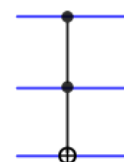
Definition(en): Quanten-Schaltkreis

Quanten-Gatter werden durch kalibrierte Laser, elektrische oder magnetische Felder oder Mikrowellen realisiert

dabei muss die Erzeugung von Fehlern möglichst gering sein

notwendig ist ein universelles Set an Gattern

ein universelles Gatter, dass alle Operationen realisieren kann ist das TOFFOLI-Gatter



## Definition(en): Quanten-Gatter

Ein Quanten-Gatter ist eine Steuereinrichtung, welche die Wechselwirkungen von Qubit's untereinander oder mit der Umgebung beeinflusst.

Simulation eines Quanten-Schaltkreises auf: <https://algassert.com/quirk>

### Aufruf-Link zur Simulation:

[https://algassert.com/quirk#circuit=\[%22cols%22:\[\[%22X%22,1,%22%E2%80%A2%22\],\[1,1,%22H%22\],\[%22Measure%22,1,%22Measure%22\],\[%22%E2%80%A2%22,%22Y%22\],\[1,%22Z%22,%22%E2%80%A2%22\],%22init%22:\[0,1,1\]\]](https://algassert.com/quirk#circuit=[%22cols%22:[[%22X%22,1,%22%E2%80%A2%22],[1,1,%22H%22],[%22Measure%22,1,%22Measure%22],[%22%E2%80%A2%22,%22Y%22],[1,%22Z%22,%22%E2%80%A2%22],%22init%22:[0,1,1]])

### interessante Links:

<https://algassert.com/quirk> Simulation von Quanten-Schaltkreisen

### 1.4.3. Quanten-Register

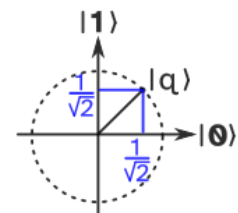
mehrere Quanten-Bit's (Qubit's) bilden ein Register  
neben der Betrachtung des Verhalten's der einzelnen Qubit's muss nun auch noch mit beachtet werden, wie die Qubit's untereinander verbunden / verschränkt sind

Quanten-Register aus 3 Qubit's hat die Gestalt:

Einzel- Wahrschein- lichkeit / Zustände	Quadrate der Einzel- Wahrschein- lichkeiten
$\alpha_0 *  000\rangle$	$ \alpha_0 ^2$
$+ \alpha_1 *  001\rangle$	$+  \alpha_1 ^2$
$+ \alpha_2 *  010\rangle$	$+  \alpha_2 ^2$
$+ \alpha_3 *  011\rangle$	$+  \alpha_3 ^2$
$+ \alpha_4 *  100\rangle$	$+  \alpha_4 ^2$
$+ \alpha_5 *  101\rangle$	$+  \alpha_5 ^2$
$+ \alpha_6 *  110\rangle$	$+  \alpha_6 ^2$
$+ \alpha_7 *  111\rangle$	$+  \alpha_7 ^2$
	<b>= 1,0</b>

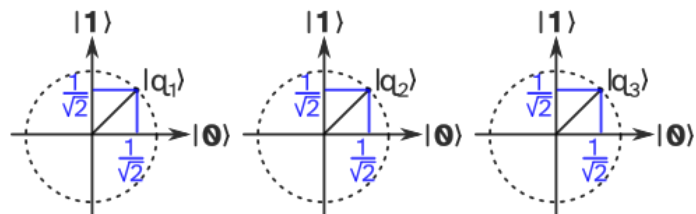
ausgehend vom klassischen Münzwurf mit 3 Münzen gibt es 8 Zustände (000) bis (111) (wo bei Kopf z.B. 0 bedeutet und 1 die Zahl)  
praktisch treten alle 8 Zustände gleich-wahrscheinlich auf (praktisch 1/8); gültig nur für ungezinkte / faire Münzen  
bei gezinkten Münzen ist die Verteilung leicht verändert, also ungleich wahrscheinlich  
außer für den Fall, dass Münzen immer das gleich Ergebnis liefern, ist es nicht möglich alle Münzen (zusammen) so zu präparieren, dass immer ein bestimmtes Ergebnis auftaucht  
jede Münze hat immer noch seine eigene Wahrscheinlichkeit und die ist unabhängig von den Ergebnisse der anderen Münzen

wenn sich ein Qubit fair ist, dann wäre  $\alpha$  und  $\beta$  jeweils  $\frac{1}{\sqrt{2}}$



drei verschränkte faire Qubit's

werden die drei Qubit's zu einem zeitpunkt gemessen, dann erhalten wir einen der Zustände  $|000\rangle, |001\rangle \dots |111\rangle$ .



daraus ergibt sich für jeden Zustand eine Wahrscheinlichkeit für a bei jedem Qubit mit  $\frac{1}{\sqrt{8}}$

damit ergibt sich als Zustand des Quanten-Register's

Einzel- Wahrschein-	Quadrate der Einzel-
------------------------	-------------------------

Berechnung Quadrate und Summe ???

Wahrscheinlichkeit / Zustände	Wahrscheinlichkeiten
$\frac{1}{\sqrt{8}} *  000\rangle$	$ \frac{1}{\sqrt{8}} ^2$
$+ \frac{1}{\sqrt{8}} *  001\rangle$	$+  \frac{1}{\sqrt{8}} ^2$
$+ \frac{1}{\sqrt{8}} *  010\rangle$	$+  \frac{1}{\sqrt{8}} ^2$
$+ \frac{1}{\sqrt{8}} *  011\rangle$	$+  \frac{1}{\sqrt{8}} ^2$
$+ \frac{1}{\sqrt{8}} *  100\rangle$	$+  \frac{1}{\sqrt{8}} ^2$
$+ \frac{1}{\sqrt{8}} *  101\rangle$	$+  \frac{1}{\sqrt{8}} ^2$
$+ \frac{1}{\sqrt{8}} *  110\rangle$	$+  \frac{1}{\sqrt{8}} ^2$
$+ \frac{1}{\sqrt{8}} *  111\rangle$	$+  \frac{1}{\sqrt{8}} ^2$
	<b>= 1,0</b>

### Definition(en): Quanten-Register

besteht das Quanten-System aus 2 Qubit's dann schreiben wir das z.B. so auf.

$$|\Psi\rangle = |\varphi\rangle_1 |\varphi\rangle_2 = |\varphi\rangle_1 \otimes |\varphi\rangle_2$$

ist z.B.  $|\varphi\rangle_1 = \alpha_1|0\rangle + \beta_1|1\rangle$  und  $|\varphi\rangle_2 = \alpha_2|0\rangle + \beta_2|1\rangle$  dann ergibt sich:

$$\begin{aligned} |\varphi\rangle_1 \otimes |\varphi\rangle_2 &= \alpha_1|0\rangle + \beta_1|1\rangle \otimes \alpha_2|0\rangle + \beta_2|1\rangle \\ &= \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|10\rangle + \beta_1\beta_2|11\rangle \end{aligned}$$

können wir ein 2-Qubit-Quanten-System wie oben schreiben – die Verknüpfung der einzelnen Qubit's erfolgt über das tensor-Produkt, dann spricht man von einem **separierbaren** bzw. **unverschränkten** System

sollte dies nicht so sein, dann sind die Qubit's (miteinander) **verschränkt** (bzw. nicht-separierbar) und lassen sich in den BELL-Zuständen beschreiben:

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) & |\Phi^-\rangle &= \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle) \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle) & |\Psi^-\rangle &= \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle) \end{aligned}$$

### Definition(en): Basis-Zustand

Die Basis-Zustände (eines Register's) sind die möglichen Mess-Ergebnisse für ein Quanten-Register.

In den von uns betrachteten 1-Qubit-Quanten-Systemen sind die Basis-Zustände  $|0\rangle$  und  $|1\rangle$ .

### Definition(en): Amplitude

Eine Amplitude ist ein  $\alpha$ -Wert für einen Register-Zustand / Basis-Zustand.

### Definition(en): Superposition

Eine Superposition ist die Situation, wenn zwei Amplituden ( $\alpha$ -Werte von Basis-Zuständen) nicht 0 sind.

Es ist in diesem Fall nicht eindeutig, welchen Zustand die Qubit's letztendlich einnehmen.

ist ein Qubit mit einer negativen Amplitude versehen (hier  $q_7$ )

Einzel-  
Wahrschein-  
lichkeit /  
Zustände

$$\begin{aligned} & \frac{1}{\sqrt{8}} * |000\rangle \\ - & \frac{1}{\sqrt{8}} * |001\rangle \\ + & \frac{1}{\sqrt{8}} * |010\rangle \\ - & \frac{1}{\sqrt{8}} * |011\rangle \\ + & \frac{1}{\sqrt{8}} * |100\rangle \\ - & \frac{1}{\sqrt{8}} * |101\rangle \\ + & \frac{1}{\sqrt{8}} * |110\rangle \\ - & \frac{1}{\sqrt{8}} * |111\rangle \end{aligned}$$

Quadrate der  
Einzel-  
Wahrschein-  
lichkeiten

$$\begin{aligned} & |\alpha_0|^2 \\ + & |\alpha_1|^2 \\ + & |\alpha_2|^2 \\ + & |\alpha_3|^2 \\ + & |\alpha_4|^2 \\ + & |\alpha_5|^2 \\ + & |\alpha_6|^2 \\ + & |\alpha_7|^2 \\ \hline & = 1,0 \end{aligned}$$

für verschränkte Qubits ergeben sich

Einzel-  
Wahrschein-  
lichkeit /  
Zustände

$$\begin{aligned} & \frac{1}{\sqrt{2}} * |000\rangle \\ + & 0 * |001\rangle \\ + & 0 * |010\rangle \\ + & 0 * |011\rangle \\ + & 0 * |100\rangle \\ + & 0 * |101\rangle \\ + & 0 * |110\rangle \\ + & \frac{1}{\sqrt{2}} * |111\rangle \end{aligned}$$

Quadrate der  
Einzel-  
Wahrschein-  
lichkeiten

$$\begin{aligned} & |\alpha_0|^2 \\ + & |\alpha_1|^2 \\ + & |\alpha_2|^2 \\ + & |\alpha_3|^2 \\ + & |\alpha_4|^2 \\ + & |\alpha_5|^2 \\ + & |\alpha_6|^2 \\ + & |\alpha_7|^2 \\ \hline & = 1,0 \end{aligned}$$

### 1.4.4.1. Berechnen von Quanten-Registern

ausgehend von 3 Qubit's mit

$$\begin{aligned} |q_1\rangle &= \frac{1}{\sqrt{2}} * |0\rangle + \frac{1}{\sqrt{2}} * |1\rangle \\ |q_2\rangle &= \frac{1}{\sqrt{2}} * |0\rangle + \frac{1}{\sqrt{2}} * |1\rangle \\ |q_3\rangle &= \frac{1}{\sqrt{2}} * |0\rangle + \frac{1}{\sqrt{2}} * |1\rangle \end{aligned}$$

?  $|q_1, q_2, q_3\rangle$

über Tensor-Produkt

$$|q_1, q_2, q_3\rangle = |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle$$

Einzel-  
Wahrschein-  
lichkeit /  
Zustände

Quadrate der  
Einzel-  
Wahrschein-  
lichkeiten

$\frac{1}{\sqrt{8}} *  000\rangle$	$ \alpha_0 ^2$
$+ \frac{1}{\sqrt{8}} *  001\rangle$	$+  \alpha_1 ^2$
$+ \frac{1}{\sqrt{8}} *  010\rangle$	$+  \alpha_2 ^2$
$+ \frac{1}{\sqrt{8}} *  011\rangle$	$+  \alpha_3 ^2$
$+ \frac{1}{\sqrt{8}} *  100\rangle$	$+  \alpha_4 ^2$
$+ \frac{1}{\sqrt{8}} *  101\rangle$	$+  \alpha_5 ^2$
$+ \frac{1}{\sqrt{8}} *  110\rangle$	$+  \alpha_6 ^2$
$+ \frac{1}{\sqrt{8}} *  111\rangle$	$+  \alpha_7 ^2$
	<b>= 1,0</b>

$$\begin{aligned} |q_1, q_2, q_3\rangle &= \left( \frac{1}{\sqrt{2}} * |0\rangle + \frac{1}{\sqrt{2}} * |1\rangle \right) \otimes \left( \frac{1}{\sqrt{2}} * |0\rangle + \frac{1}{\sqrt{2}} * |1\rangle \right) \otimes \left( \frac{1}{\sqrt{2}} * |0\rangle + \frac{1}{\sqrt{2}} * |1\rangle \right) \\ &= \frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \\ &\quad + \frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |1\rangle \\ &\quad + \frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |1\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \\ &\quad + \frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |1\rangle \otimes \frac{1}{\sqrt{2}} * |1\rangle \\ &\quad + \frac{1}{\sqrt{2}} * |1\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \\ &\quad + \frac{1}{\sqrt{2}} * |1\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |1\rangle \\ &\quad + \frac{1}{\sqrt{2}} * |1\rangle \otimes \frac{1}{\sqrt{2}} * |1\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \\ &\quad + \frac{1}{\sqrt{2}} * |1\rangle \otimes \frac{1}{\sqrt{2}} * |1\rangle \otimes \frac{1}{\sqrt{2}} * |1\rangle \end{aligned}$$

Vereinfachung

$$\frac{1}{\sqrt{2}} * |0\rangle \otimes \frac{1}{\sqrt{2}} * |0\rangle \otimes \left( \frac{1}{\sqrt{2}} * |0\rangle \right) = \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} * |0\rangle \otimes |0\rangle \otimes |0\rangle = \frac{1}{\sqrt{8}} * |000\rangle$$

ergibt dann:

$$\begin{aligned} |q_1, q_2, q_3\rangle &= \frac{1}{\sqrt{8}} * |000\rangle \\ &= + \frac{1}{\sqrt{8}} * |001\rangle \\ &= + \frac{1}{\sqrt{8}} * |010\rangle \\ &= + \frac{1}{\sqrt{8}} * |011\rangle \end{aligned}$$

$$\begin{aligned}
&= + \frac{1}{\sqrt{8}} * |100\rangle \\
&= + \frac{1}{\sqrt{8}} * |101\rangle \\
&= + \frac{1}{\sqrt{8}} * |110\rangle \\
&= + \frac{1}{\sqrt{8}} * |111\rangle
\end{aligned}$$

gilt ja nur für faire Qubit's; allg. gilt dann:

$$\begin{aligned}
|q_1, q_2, q_3\rangle &= |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle \\
&= (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle) \otimes (\alpha_3|0\rangle + \beta_3|1\rangle) \\
&= \alpha_1\alpha_2\alpha_3|000\rangle \\
&\quad + \alpha_1\alpha_2\beta_3|001\rangle \\
&\quad + \alpha_1\beta_2\alpha_3|010\rangle \\
&\quad + \alpha_1\beta_2\beta_3|011\rangle \\
&\quad + \beta_1\alpha_2\alpha_3|100\rangle \\
&\quad + \beta_1\alpha_2\beta_3|101\rangle \\
&\quad + \beta_1\beta_2\alpha_3|110\rangle \\
&\quad + \beta_1\beta_2\beta_3|111\rangle
\end{aligned}$$

Man nennt Quanten-Register, die sich als Tensor-Produkt der einzelnen Qubit's beschreiben lassen, **separierbar** bzw. **unverschränkt**.

Mit anderen Worten: Sind die einzelnen Qubit's eines Quanten-Register, unverschränkt / separierbar / unabhängig voneinander, dann lässt sich der Zustand des Quanten-Register's über das Tensor-Produkt der einzelnen Qubit's berechnen.

Sind die Qubit's dagegen **verschränkt** bzw. **nicht separierbar**, dann gilt:

$$|q_1, q_2, q_3\rangle = \frac{1}{\sqrt{2}} |000\rangle + \frac{1}{\sqrt{2}} |111\rangle$$

Aus den obigen Formeln leitet sich auch die Matrix-Schreibweise von Quanten-Registern ab: die beiden Quantenbit-Zustände werden in einem Basis-Vektor notiert

$$\text{durch Basis-Vektoren: } |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \begin{matrix} |0\rangle & (?) \\ |1\rangle & (?) \end{matrix}$$

$$\text{allgemein: } |q\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

die Möglichkeiten (Register-Zustände) werden nacheinander Zeilen-weise notiert.

Die Matrix enthält dann jeweils genutzten Register-Zustand (mit 1 gekennzeichnet)

Amplituden werden vor den Vektor notiert

für ein Register mit 2 Qubit's ergibt sich dann:

$$\begin{matrix} |00\rangle & (?) \\ |01\rangle & ? \\ |10\rangle & ? \\ |11\rangle & (?) \end{matrix} \begin{pmatrix} ? \\ ? \\ ? \\ ? \end{pmatrix}$$

im Folgenden zeigen wir meist mehrere Darstellungs- bzw. Visualisierungs-Formen für die Register und Gatter

i.A. reicht es, wenn man sich auf eine Form einschiesst, begleitend sind die Formeln aber sicher immer eine recht verständliche und universelle Form



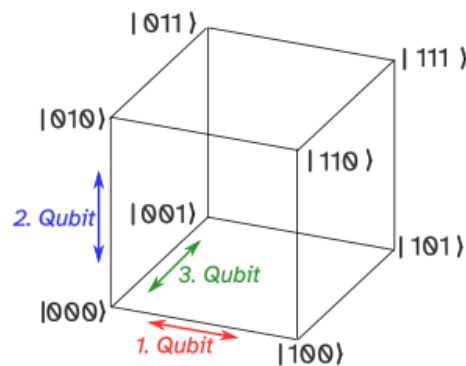
### 1.4.4.2. Veranschaulichen / Illustration von Quanten-Registern

Das folgende Quanten-Register soll etwas besser illustriert werden.

$$\begin{aligned}
 |q_1, q_2, q_3\rangle &= \sqrt{\frac{2}{16}} * |000\rangle \\
 &= + \sqrt{\frac{0}{16}} * |001\rangle \\
 &= - \sqrt{\frac{2}{16}} * |010\rangle \\
 &= - \sqrt{\frac{3}{16}} * |011\rangle \\
 &= - \sqrt{\frac{1}{16}} * |100\rangle \\
 &= + \sqrt{\frac{1}{16}} * |101\rangle \\
 &= + \sqrt{\frac{4}{16}} * |110\rangle \\
 &= + \sqrt{\frac{3}{16}} * |111\rangle
 \end{aligned}$$

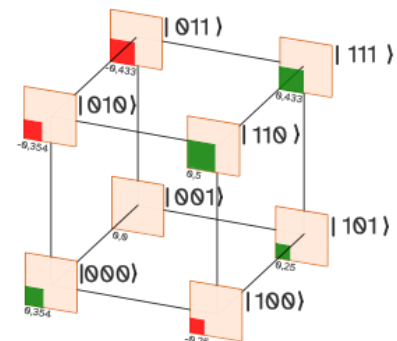
Wir bleiben dabei zuerst einmal bei 3 Qubits.

Zur Veranschaulichung benutzen wir das unten gezeichnete Würfel-Modell (nach JUST). Die Qubit's stellen die Richtungen im kartesischen Koordinaten-System dar.



Dadurch liegen die Register-Zustände an den Ecken.

Im nächsten Schritt werden die Amplituden hinzugefügt. Dazu erhält jeder Zustand eine eigene "Anzeige-Tafel", in der die Amplitude als Fläche dargestellt ist. Eine rote Fläche bedeutet einen positiven Wert, eine grüne entsprechend eine negative. Die Summe aller Flächen muss am Ende genau 1 also einmal das "Anzeige-Quadrat" ergeben. Die kleinen Zahlenwerte unter den kleinen Quadraten sind die Amplituden.



Kleinere Quanten-Register mit einem oder zwei Qubit's ließen sich als Strecke / Kante bzw. als Fläche darstellen.

Will man dagegen ein Quanten-Register mit vier Qubit's veranschaulichen, dann stoßen wir mit unseren üblichen Dimensionen an die Grenze. Eine Hilfs-Lösung ist ein sogenannter Hyper-Würfel – also einem Würfel in einem Würfel (s.a. Abb. weiter hinten).

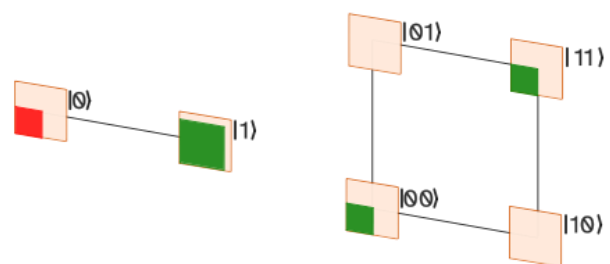
Mit Hilfe dieses Darstellungs-Verfahren's können wir auch gut erkennen, ob Qubit's verschränkt oder unverschränkt sind.

Vergleicht man ein unverschränktes und ein verschränktes Quanten-Register mit jeweils zwei fairen Qubit's, dann ergeben sich völlig unterschiedliche Schmata.

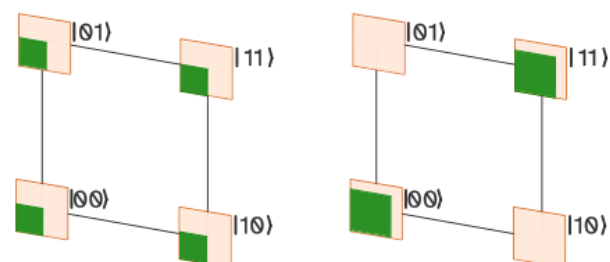
Links das unverschränkte System:

$$\left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle\right)$$

und rechts das entsprechende ver-

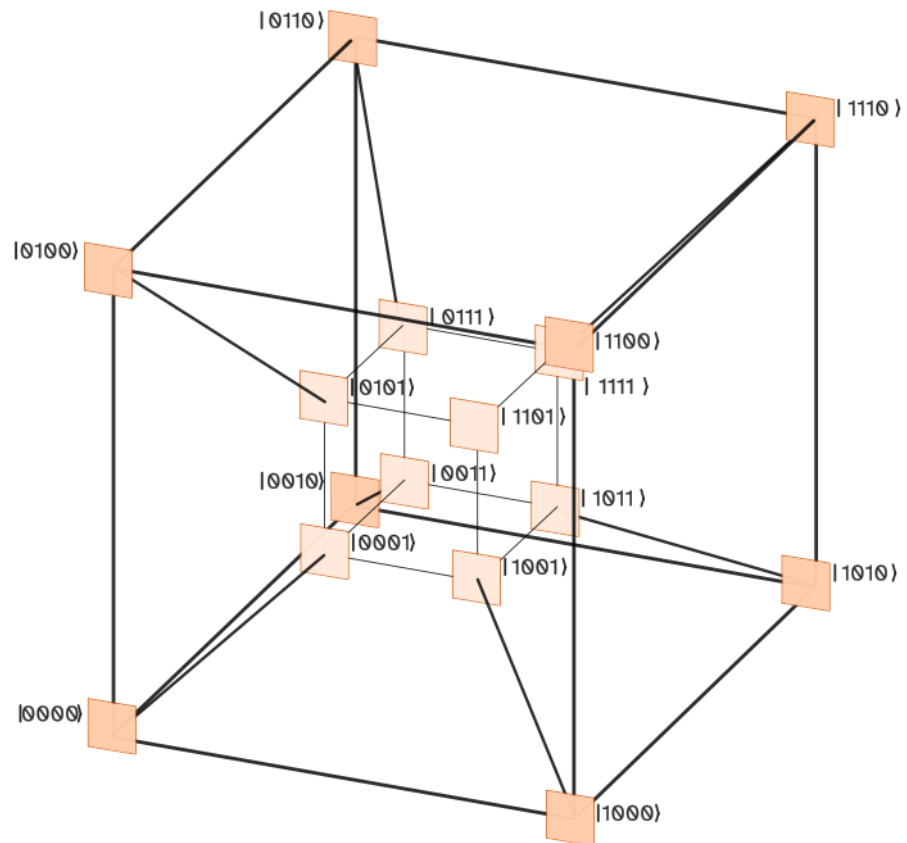


1-Qubit-Register  $\left(-\frac{1}{2} |0\rangle + \frac{\sqrt{3}}{2} |1\rangle\right)$   
 und 2-Qubit-Register  $\left(\frac{1}{\sqrt{2}} |00\rangle + |11\rangle\right)$



schränkte System:

$$\frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$$



Modell für ein Quanten-Register mit 4 Qubit's (Darstellung als Hyper-Würfel)

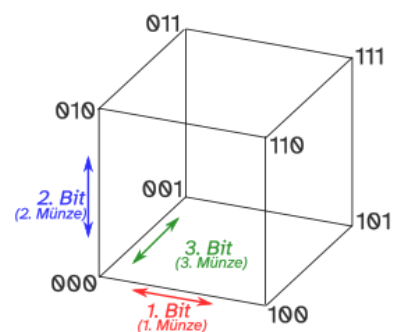
### 1.4.4.3. Veranschaulichung des Messen's in Quanten-Registern

Erst mit der Messung nehmen Qubit's einen bestimmten Wert ein. Es kann aber auch sein, dass die Messung des einen Qubit's auch ein anderes Qubit mit beeinflusst. Das ist immer dann so, wenn die beiden Qubit's miteinander verschränkt sind.

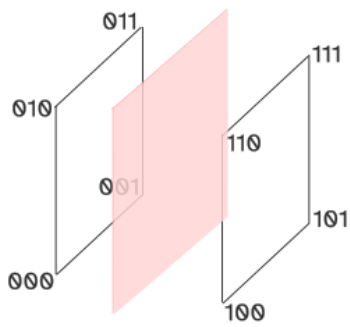
Rück-Griff auf Messung / Beobachtung in einem fairen Münzwurf-System. Jeder der Ergebnis-Zustände ist gleich groß. Wieder angenommen Kopf wird durch 0 und Zahl durch eine 1 repräsentiert, dann ergibt sich als Modell ein Würfel mit den Zuständen an den Ecken. Auch hier ist die Zustands-Verteilung für eine Münze wie oben. Links/rechts für die erste Münze, hoch/runter für die zweite und vorn/hinten für die dritte Münze. Praktisch sind wir auf der Ebene von Bit's.

Bei einem Münzwurf-Versuch mit fairen Münzen soll das Ergebnis noch nicht sichtbar sein. In diesem Fall sind alle Zustände (Würfel-Ecken) möglich.

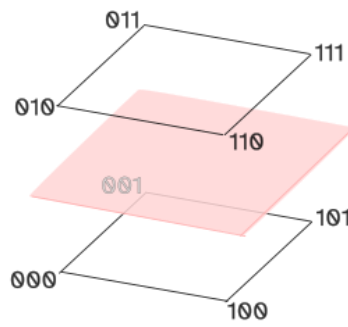
Deckt man nun eine der Münzen auf, dann teilt sich der Ergebnis-Würfel quasi in zwei Flächen. Dabei ist nur noch eine gültig. Zb.B. nach dem Aufdecken der 1. Münze entweder die linke oder die rechte Fläche (s.a. linke Abb.).



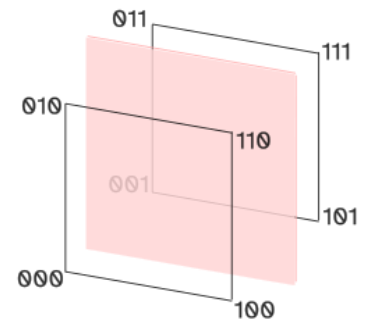
Praktisch ist es auch egal, welche der Münzen man zuerst aufdeckt, die Anzahl der nun gültigen Möglichkeiten ist immer vier.



nur 1. Münze aufgedeckt



nur 2. Münze aufgedeckt

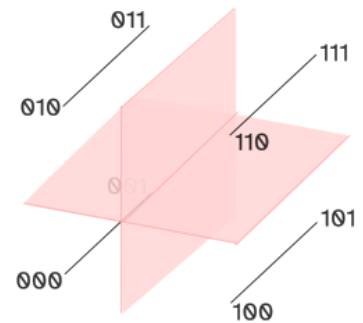


nur 3. Münze aufgedeckt

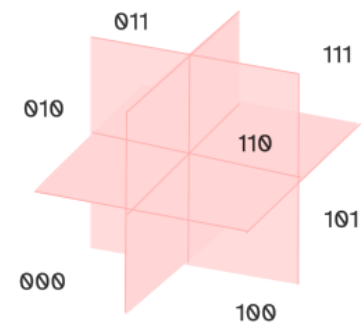
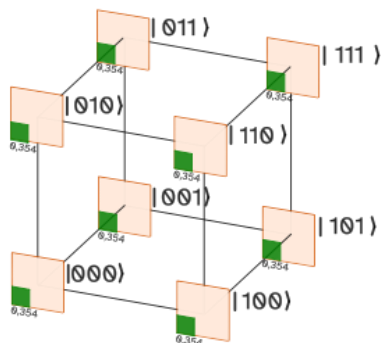
Mit dem Aufdecken einer zweiten Münze verringert sich die Menge der möglichen Ergebnisse weiter. Wir befinden uns jetzt im Bereich einer der vier Kanten. Es bleiben somit nur noch zwei mögliche Zustände.

Durch das Aufdecken der zwei Münzen wissen wir schon, wo wir uns im Würfel befinden.

Wurden alle Münzen aufgedeckt, dann ist das Ergebnis eindeutig. Wir befinden uns in einer der Ecken des Ergebnis-Würfels.



Das gleiche Verfahren können wir nun auch bei den Quanten-Registern anwenden.

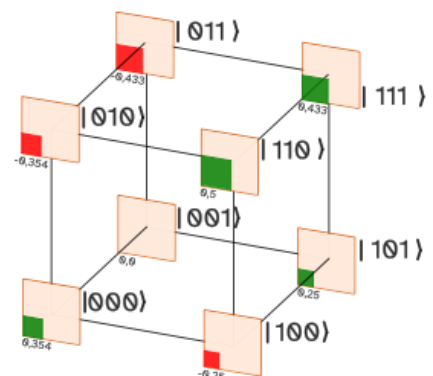


Um die besonderen Effekte zu zeigen, nutzen wir nicht ein faires Quanten-Register (s. Abb. hier drüber), sondern ein spezielles mit abweichenden Amplituden (s. Abb. rechts).

Wie sich die Messung an einem fairem System reüräsentiert, kann man dann leicht nachvollziehen. Das ist eine schöne Übung.

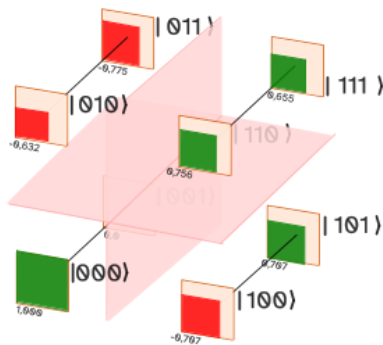
Mit dem Messen des ersten Qubit's teilen wir den Würfel genauso, wie oben bei den Münzen.

Wieder bekommen wir eingegrenzte Ergebnis-Welten. Es tritt nun aber ein besonderer Effekt ein. Die Summe der Amplituden-Quadrate muss jetzt aber innerhalb des noch quantifizierten Bereiches gleich 1 sein.

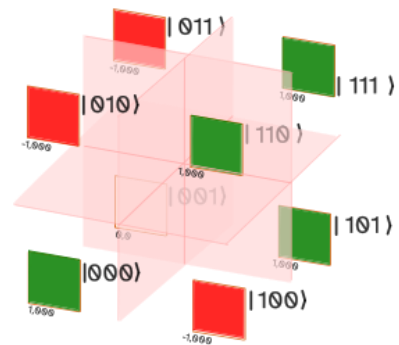




nur 1. Qubit gemessen

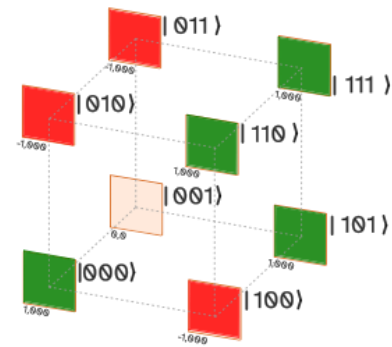


1. und 2. Qubit gemessen



alle 3 Qubit gemessen

Die Einzell-Amplituden müssen dementsprechend angepasst (normiert) werden.  
Verzichtet man am Ende noch auf die Schnittflächen, dann wird das Endergebnis nach allen drei Messungen noch besser sichtbar.

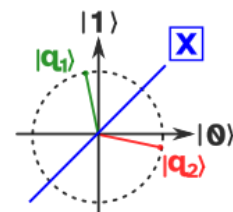


### 1.4.4.4. verschiedene Quanten-Gatter in Quanten-Registern

#### PAULI-X-Gatter, X-Gatter

1-Qubit-Gatter  
tauscht die Amplituden eines Qubit's

in der Vektor-Darstellung an einem Kreis entspricht die Funktion des PAULI-X-Gatter's einer Spiegelung des Vektor's an einer 45°-Geraden (Fkt.:  $y = -x$ )

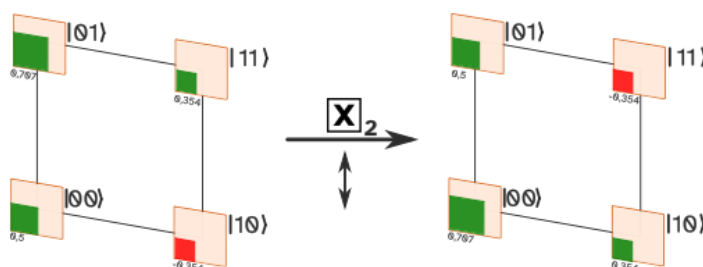
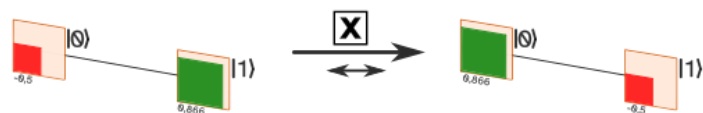


PAULI-X-Gatter als Vektoren-Spiegelung an einer 45°-Geraden

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{X} \beta|0\rangle + \alpha|1\rangle$$

$$\text{also } |0\rangle \xrightarrow{X} |1\rangle$$

bzw.  $|1\rangle \xrightarrow{X} |0\rangle$   
bei Anwendung eines PAULI-X-Gatter's auf ein Qubit in einem Quanten-Register ändert die Amplituden an einer "Spiegel"-Ebene für das 2. Quanten-Bit wäre das in unserem Darstellungs-Modell die waagerechte Ebene zwischen den Qubit's



$$|q_1 q_2\rangle = \frac{1}{2} |00\rangle + \frac{1}{\sqrt{2}} |01\rangle - \frac{1}{\sqrt{8}} |10\rangle + \frac{1}{\sqrt{8}} |11\rangle$$

$X_2$

$$|q_1 q_2\rangle = \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{2} |01\rangle + \frac{1}{\sqrt{8}} |10\rangle - \frac{1}{\sqrt{8}} |11\rangle$$

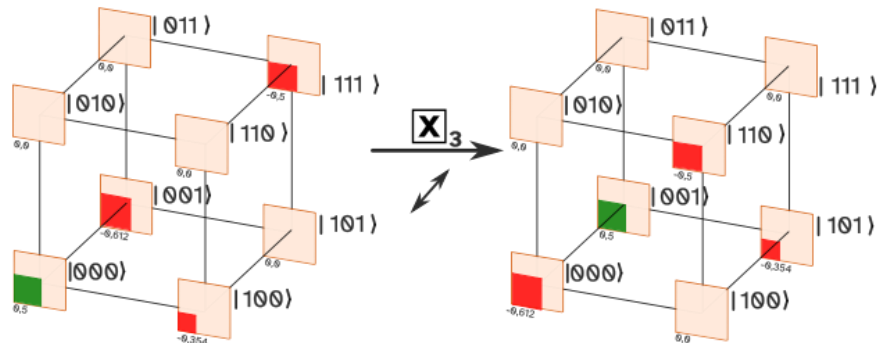
alternativ ließe sich das auch durch Tauschen der Zustände bei gleichliegenden Amplituden darstellen:

$$|q_1 q_2\rangle = \frac{1}{2} |01\rangle + \frac{1}{\sqrt{2}} |00\rangle - \frac{1}{\sqrt{8}} |11\rangle + \frac{1}{\sqrt{8}} |10\rangle$$

Dieses Prinzip lässt auch leicht auf Quanten-Register mit 3 Qubit's anwenden.

Als Beispiel soll das PAULI-X-Gatter auf das 3. Qubit angewendet werden.

Nun steht die "Spiegel"-Ebene senkrecht und ist praktisch parallel zur Betrachtungs-Vorderseite somit wird die Vorder- und die Hinter-Seite getauscht



$$|q_1 q_2 q_3\rangle = \frac{1}{2} |000\rangle - \sqrt{\frac{3}{8}} |001\rangle - \frac{1}{\sqrt{8}} |100\rangle - \frac{1}{2} |111\rangle$$

$X_3$

$$|q_1 q_2 q_3\rangle = \frac{1}{2} |001\rangle - \sqrt{\frac{3}{8}} |000\rangle - \frac{1}{\sqrt{8}} |101\rangle - \frac{1}{2} |110\rangle$$

bzw. alternativ (mit getauschten Amplituden):

$$|q_1 q_2 q_3\rangle = -\sqrt{\frac{3}{8}} |000\rangle + \frac{1}{2} |001\rangle - \frac{1}{\sqrt{8}} |101\rangle - \frac{1}{2} |110\rangle$$

**Aufgaben:**

- 1.
2. *Denken Sie sich ein 2-Qubit-Register aus! Übergeben Sie dieses einem Kurs-Partner und beauftragen Sie diesen, ein PAULI-X-Gatter auf das 1. Qubit anzuwenden! Wenn der Partner fertig ist, kontrollieren Sie gemeinsam das Ergebnis!*
3. *Notieren Sie das obige Beispiel-Quanten-Register in der vollständigen Form (mit allen Quanten-Zuständen)! Wenden Sie dann das PAULI-X-Gatter auf das 3. Qubit des Register's an und notieren Sie das Ergebnis in der gewünschten Form! Kennzeichnen Sie die Stellen, an denen das Gatter gewirkt hat!*
4. *Notieren Sie das obige Beispiel-Quanten-Register in der vollständigen oder gekürzten Form! Wenden Sie dann das PAULI-X-Gatter auf das 2. Qubit des Register an und notieren Sie das Ergebnis in der gewünschten Form! Kennzeichnen Sie die Stellen, an denen das Gatter gewirkt hat!*

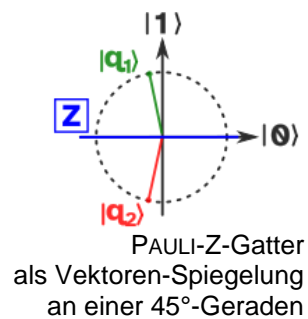
**PAULI-Y-Gatter, Y-Gatter**

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

**PAULI-Z-Gatter, Z-Gatter**

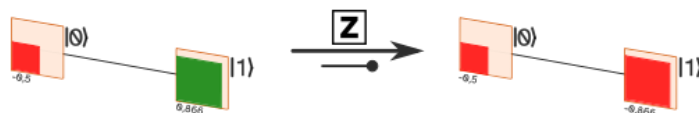
1-Qubit-Gatter  
tauscht die Amplituden eines Qubit's

bei Anwendung des PAULI-Z-Gatter's auf ein Quanten-Bit wird bei dem Qubit das Vorzeichen der Amplitude des  $|0\rangle$ -Zustand's gewechselt  
praktisch Spiegelung an der Waagerechten



$$|1\rangle \xrightarrow{Z} -|1\rangle \quad \text{oder} \quad -|1\rangle \xrightarrow{Z} |1\rangle \quad \text{sowie} \quad |0\rangle \xrightarrow{Z} |0\rangle$$

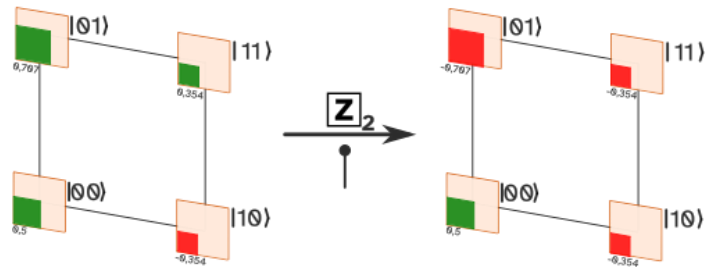
allgemein also:  
 $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle - \beta|1\rangle$



schreibt man das Beispiel mit Amplituden zu den Zuständen

$$-\frac{1}{2} |0\rangle + \frac{\sqrt{3}}{2} |1\rangle \xrightarrow{Z} -\frac{1}{2} |0\rangle - \frac{\sqrt{3}}{2} |1\rangle$$

wird ein PAULI-Z-Gatter auf ein Qubit in einem Quanten-Register angewendet (hier auf das 2. Qubit), dann hat das auch Auswirkungen auf die anderen Qubit's  
es wechselt das Vorzeichen aller  $|1\rangle$ -Zustände

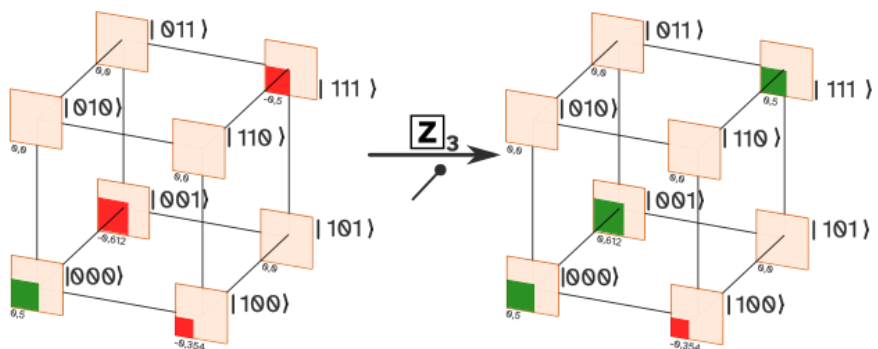


$$|q_1 q_2\rangle = \frac{1}{2} |00\rangle + \frac{1}{\sqrt{2}} |01\rangle - \frac{1}{\sqrt{8}} |10\rangle + \frac{1}{\sqrt{8}} |11\rangle$$

$Z_2$

$$|q_1 q_2\rangle = \frac{1}{2} |00\rangle - \frac{1}{2} |01\rangle - \frac{1}{\sqrt{8}} |10\rangle - \frac{1}{\sqrt{8}} |11\rangle$$

wendet man nun das PAULI-Z-Gatter auf das 3. Qubit an, dann ergibt analog



für das Beispiel-Register ergibt sich:

$$|q_1 q_2 q_3\rangle = \frac{1}{2} |000\rangle - \sqrt{\frac{3}{8}} |001\rangle - \frac{1}{\sqrt{8}} |100\rangle - \frac{1}{2} |111\rangle$$

$Z_3$

$$|q_1 q_2 q_3\rangle = \frac{1}{2} |000\rangle + \sqrt{\frac{3}{8}} |001\rangle - \frac{1}{\sqrt{8}} |100\rangle + \frac{1}{2} |111\rangle$$

was eben nur Vorzeichen-Wechseln bei den Zuständen ergibt, deren 3. Qubit den Zustand  $|1\rangle$  haben

---

**Aufgaben:**

- 1.
2. *Denken Sie sich ein 2-Qubit-Register aus! Übergeben Sie dieses einem Kurs-Partner und beauftragen Sie diesen, ein PAULI-Z-Gatter auf das 1. Qubit anzuwenden! Wenn der Partner fertig ist, kontrollieren Sie gemeinsam das Ergebnis!*
3. *Notieren Sie das obige Beispiel-Quanten-Register in der vollständigen Form (mit allen Quanten-Zuständen)! Wenden Sie dann das PAULI-Z-Gatter auf das 3. Qubit des Register's an und notieren Sie das Ergebnis in der gewünschten Form! Kennzeichnen Sie die Stellen, an denen das Gatter gewirkt hat!*
4. *Notieren Sie das obige Beispiel-Quanten-Register in der vollständigen oder gekürzten Form! Wenden Sie dann das PAULI-Z-Gatter auf das 2. Qubit des Register an und notieren Sie das Ergebnis in der gewünschten Form! Kennzeichnen Sie die Stellen, an denen das Gatter gewirkt hat!*



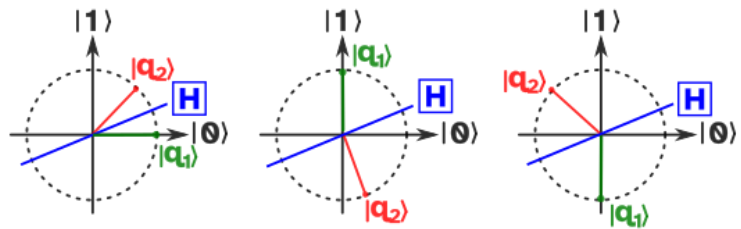
## HADAMARD-Gatter, H-Gatter

erzeugt gleich-verteilte Superposition

Spiegeln an einer 22,5°-Geraden

als Matrix:

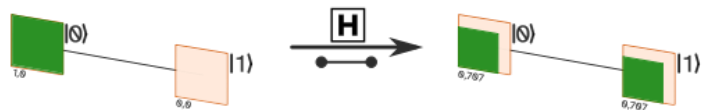
$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



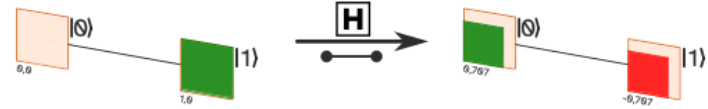
Wirkung von HADAMARD auf die Zustände  $|0\rangle$ ,  $|1\rangle$  und  $-|1\rangle$  ( $|q_1\rangle$  (grün) ... vor dem Gatter;  $|q_2\rangle$  (rot) ... nach dem Gatter)

diese Grund-Funktionen im mathematischen und dem Veranschaulichungs-Modell:

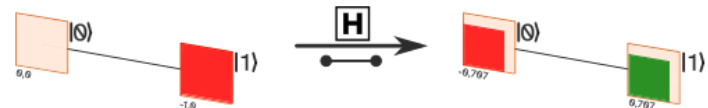
$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle = |+\rangle$$



$$|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle = |-\rangle$$



$$-|1\rangle \xrightarrow{H} -\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle = -|-\rangle$$



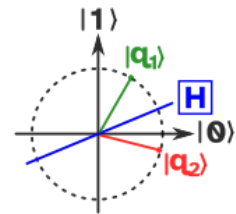
allgemein gilt:

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{H} \frac{\alpha + \beta}{\sqrt{2}} |0\rangle + \frac{\alpha - \beta}{\sqrt{2}} |1\rangle$$

**Beispiel:**

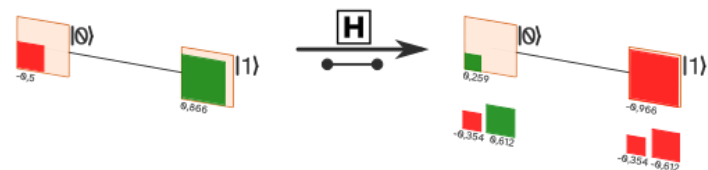
beliebiger Quanten-Zustand:

$$-\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle \xrightarrow{H} \frac{-1 + \sqrt{3}}{2\sqrt{2}} |0\rangle + \frac{-1 - \sqrt{3}}{2\sqrt{2}} |1\rangle$$



Wirkung am Flächen-Modell:

die Anwendung des H-Gatter's auf den  $|0\rangle$ - und den  $|1\rangle$ -Zustand separat sind als Nebenrechnung unter dem Ergebnis zu sehen



zu beachten ist, dass sich die Flächen addieren (nicht etwa die Amplituden)

$$|+\rangle \xrightarrow{H} |0\rangle$$

H

$|-\rangle \rightarrow |1\rangle$

wird das HADAMARD-Gatter zwei-mal hintereinander jeweils auf das Ergebnis angewendet, dann ergibt sich der Ausgangs-Zustand (Identität); die Einzel-Wirkung wird praktisch dann aufgehoben

**Anwendung des HADAMARD-Gatter's zur Erzeugung von echtem Zufall**

Quanten-Schaltkreis						
	<b>Eingang</b>		nach Gatter	Messung	<b>Ausgang</b> (Wahrscheinlichkeit)	
Register-Zustand	$ 0\rangle$	$1 *  0\rangle$ $+ 0 *  1\rangle$	$\frac{1}{\sqrt{2}} *  0\rangle$ $+ \frac{1}{\sqrt{2}} *  1\rangle$	od. $ 0\rangle$ $ 1\rangle$	50 % bzw. 0,5 50 % bzw. 0,5	

aus einem Basis-Zustand (hier: ) wird ein (nachweisbar!) echter Zufall erzeugt

derzeit schon in Hardware umgesetzt

## Nutzung des HADAMARD-Gatter's in Quanten-Registern

z.B. Quanten-Register mit 3 Qubit's (nicht verschränkt)

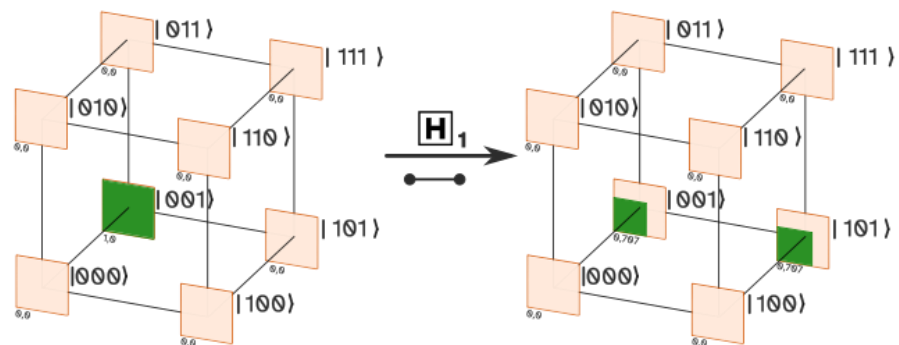
für Basis-Zustand  $|001\rangle$  lässt sich auch schreiben  $|0\rangle \otimes |01\rangle$  (od. einfacher:  $|0\rangle * |01\rangle$ )

$$\begin{aligned} H|001\rangle &= H|0\rangle * |01\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) * |01\rangle \\ &= \frac{1}{\sqrt{2}}|001\rangle + \frac{1}{\sqrt{2}}|101\rangle \end{aligned}$$

bzw. anders gerechnet durch direktes Einsetzen:

$$|001\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}|001\rangle + \frac{1}{\sqrt{2}}|101\rangle$$

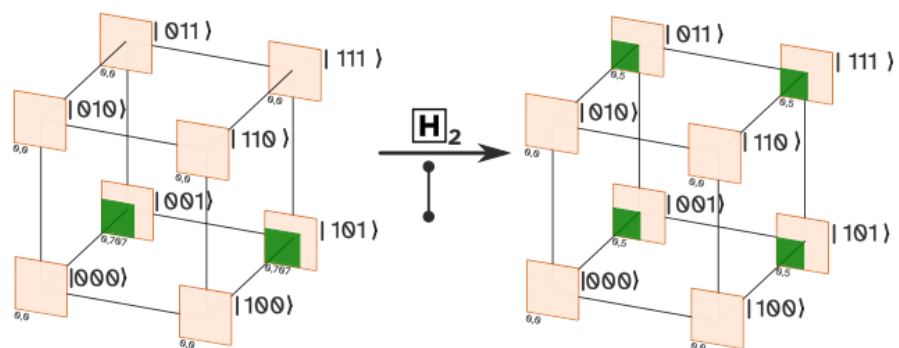
graphisch im Würfel-Modell könnte das so aussehen



in der Folge soll nun HADAMARD auf das 2. Quanten-Bit angewendet werden  
rechnerisch ergibt sich

$$H\left(\frac{1}{\sqrt{2}}|001\rangle + \frac{1}{\sqrt{2}}|101\rangle\right) = \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}}|011\rangle + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}}|101\rangle + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}}|111\rangle$$

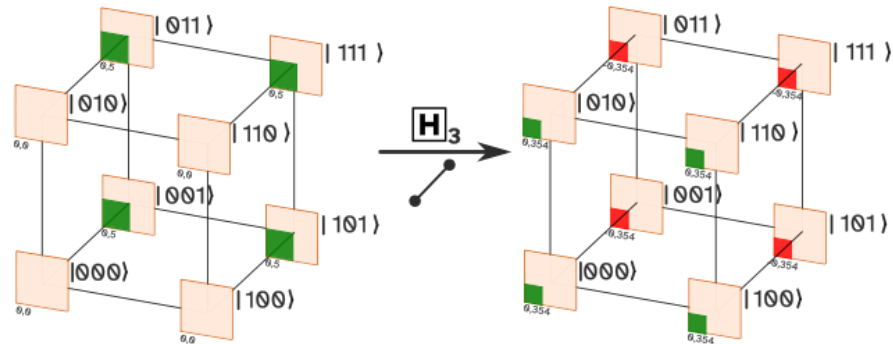
und der graphischen Darstellung im Würfel die Amplitude wurde nun quasi durch "Schütteln" gleichmäßig auf die obere und untere Ebene verteilt allerdings bleibt die Vorderseite leer, da hier ja bisher keine Amplituden vorhanden sind



im letzten Schritt wenden wir ein HADAMARD-Gatter auf das 1. Quanten-Bit an  
 Nun ist allerdings der  $|0\rangle$ -Zustand in der vorderen Ebene tragend

$$H \left( \frac{1}{2} |001\rangle + \frac{1}{2} |011\rangle + \frac{1}{2} |101\rangle + \frac{1}{2} |111\rangle \right) = \frac{1}{\sqrt{8}} |000\rangle - \frac{1}{\sqrt{8}} |001\rangle + \frac{1}{\sqrt{8}} |010\rangle - \frac{1}{\sqrt{8}} |011\rangle + \frac{1}{\sqrt{8}} |100\rangle - \frac{1}{\sqrt{8}} |101\rangle + \frac{1}{\sqrt{8}} |110\rangle - \frac{1}{\sqrt{8}} |111\rangle$$

das Würfel-Modell macht die negierten Anteile auf der Rück-Seite (entspricht ja den negativen Amplituden bei den Basis-Zuständen) schön deutlich



alle Qubit's sind jeweils einzeln geblieben, sie sind also nicht verschränkt

wir erhalten eine Superposition aus allen Basis-Zuständen

Welche Vorzeichen wir bei den Amplituden erhalten, ist nur vom Ausgangs-Zustand abhängig. Die Wahrscheinlichkeit der einzelnen Basis-Zustände ist gleichverteilt.

### Aufgaben:

1. Überlegen Sie sich, was sich verändert, wenn man HADAMARD – wie oben, aber ausgehend vom Zustand  $|101\rangle$  anwendet!
2. Wenden Sie HADAMARD nach und nach auf das 3-Qubit-Register  $|000\rangle$  an! (Entscheiden Sie sich für ein Lösungs-Verfahren (graphisch oder rechnerisch)!) **für die gehobene Anspruchsebene:**

**für die gehobene Anspruchsebene:**

3. Wenden Sie HADAMARD nach und nach auf das 3-Qubit-Register  $\frac{1}{\sqrt{2}} |000\rangle - \frac{1}{\sqrt{2}} |001\rangle$  an! (Entscheiden Sie sich für ein Lösungs-Verfahren (graphisch oder rechnerisch)!) **für die gehobene Anspruchsebene:**

## controlled NOT-Gatter, CNOT-Gatter

gesprochen sii-nott

gesteuertes Verneinen

ist ein 2 Quanten-Gatter (1 Qubit ist das Ziel-Qubit (target) und 1 Steuer-Qubit (control))

Funktion: wenn das Steuer-Qubit  $|1\rangle$ , dann wird beim Ziel-Qubit die Amplitude negiert bzw. der Basis-Zustand getauscht

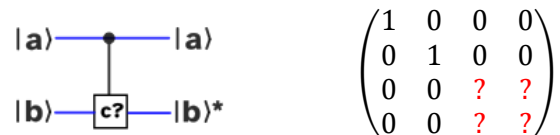
zählt zu den universellen Gattern, mit diesen lassen sich praktisch alle Funktionen / anderen Gatter nachbauen / ersetzen

es gehört zu den gesteuerten Gattern, diese

haben die nebenstehende Gatter-Struktur

und lassen sich mit einer Matrix beschreiben

der freie Bereich wird durch die spezielle Funktion des Gatters bestimmt



beim CNOT-Gatter würde die Register-Struktur und die Matrix dann so aussehen:



die besser lesbare und wohl auch verständlichere Form als mathematischer Ausdruck lautet:

$$|q_1 q_2\rangle = -\frac{1}{\sqrt{8}} |00\rangle + \frac{1}{\sqrt{2}} |01\rangle + \frac{1}{2} |10\rangle - \frac{1}{\sqrt{8}} |11\rangle$$

**CNOT<sub>12</sub>**

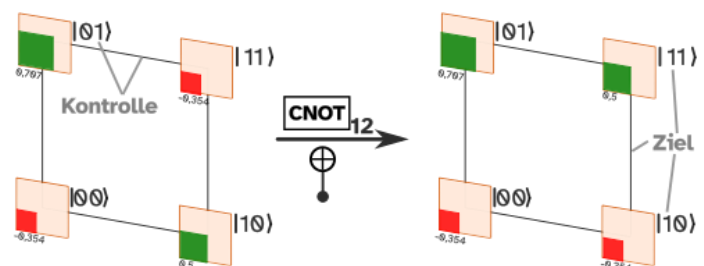
$$|q_1 q_2\rangle = -\frac{1}{\sqrt{8}} |00\rangle + \frac{1}{\sqrt{2}} |01\rangle - \frac{1}{\sqrt{8}} |10\rangle + \frac{1}{2} |11\rangle$$

bzw. durch Tausch der Zustände (im Ausdruck)

$$|q_1 q_2\rangle = -\frac{1}{\sqrt{8}} |00\rangle + \frac{1}{\sqrt{2}} |01\rangle + \frac{1}{2} |11\rangle - \frac{1}{\sqrt{8}} |10\rangle$$

die Veranschaulichung im graphischen Modell zeigt die Wirkung eben nur beim 2. Qubit (vertikale Kante) – also hoch–runter–Richtung – wenn das 1. Qubit (horizontale Kante) Eins ist

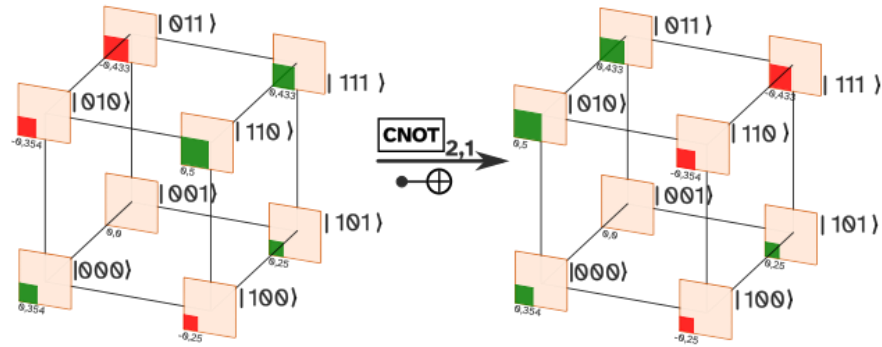
Das trifft somit nur die rechte Kante, an der eben die Amplituden getauscht werden



$$|q_1 q_2\rangle = \frac{1}{2} |01\rangle + \frac{1}{\sqrt{2}} |00\rangle - \frac{1}{\sqrt{8}} |11\rangle + \frac{1}{\sqrt{8}} |10\rangle$$

für ein Quanten-Register mit 3 Qubit's ergibt sich entsprechendes Bild

Hier wählen wir mal ein CNOT-Gatter auf dem 1. Qubit, welches durch das 2. Qubit gesteuert wird.



mathematisch ergibt sich:

$$|q_1 q_2 q_3\rangle = \frac{1}{\sqrt{8}} |000\rangle + 0 |001\rangle - \frac{1}{\sqrt{8}} |010\rangle - \frac{\sqrt{3}}{4} |011\rangle - \frac{1}{4} |100\rangle + \frac{1}{4} |101\rangle + \frac{1}{2} |110\rangle + \frac{\sqrt{3}}{4} |111\rangle$$

**CNOT<sub>2,1</sub>**

$$|q_1 q_2 q_3\rangle = \frac{1}{\sqrt{8}} |000\rangle + 0 |001\rangle + \frac{1}{2} |010\rangle + \frac{\sqrt{3}}{4} |011\rangle - \frac{1}{4} |100\rangle + \frac{1}{4} |101\rangle - \frac{1}{\sqrt{8}} |110\rangle - \frac{\sqrt{3}}{4} |111\rangle$$

wo 2. Qubit |1> ist (also oben), werden die Amplituden mit unterschiedlichem 1. Qubit (also entlang der Links-Rechts-Kanten) getauscht die vollzogenen Tausche sind also:

$$\begin{aligned} \alpha_2 |010\rangle &\rightarrow \alpha_2 |110\rangle \\ \alpha_7 |111\rangle &\rightarrow \alpha_7 |011\rangle \end{aligned}$$

$$\alpha_3 |011\rangle \rightarrow \alpha_3 |111\rangle$$

**Kontroll-Qubit**

$$\alpha_6 |110\rangle \rightarrow \alpha_6 |010\rangle$$

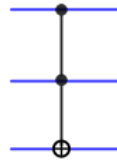
**Ziel-Qubit**

### Aufgaben:

1. Zeigen Sie die Wirkung eines CNOT<sub>1,3</sub>-Gatter's auf das obige Beispiel-Quanten-Register!
2. Wenden Sie CNOT<sub>3,1</sub> auf das obige Beispiel-Qubit-Register an! (Entscheiden Sie sich für ein Lösungs-Verfahren (graphisch oder rechnerisch)!)

## TOFFOLI-Gatter

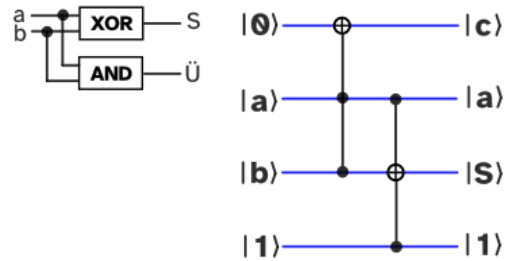
universelles Gatter, aus dem sich sehr viele Funktionen zusammenstellen lassen



$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$T = (a \wedge b) \oplus c$$

aus 2 TOFFOLI-Gattern kann ein Halb-Addierer zusammengestellt werden



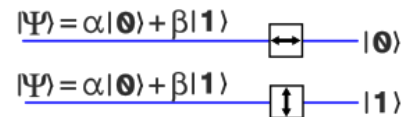
## Aufgaben:

1. Informieren Sie sich, wie das Block-Bild eines Voll-Addierers aussieht!
2. Stellen Sie einen Voll-Addierer aus TOFFOLI-Gattern zusammen!

## weitere Objekte in Quanten-Schaltkreisen

Da wir ja bei den Quanten vorrangig von schwingenden Photonen in einem Licht-Computer ausgehen, kommen auch Polarisations-Filter als Mess-Einrichtungen in den Schaltkreisen vor.

Sie prüfen, ob ein Qubit in einem bestimmten Schwingungs-Zustand ist. Passiert ein beliebiges Qubit  $|q\rangle = \alpha|0\rangle + \beta|1\rangle$  ein Polarisations-Filter für die waagerechte Schwingungs-Ebene, dann liegt das Qubit nun im Zustand  $|0\rangle$  vor. Prüft man dagegen mit einem senkrechten Polarisations-

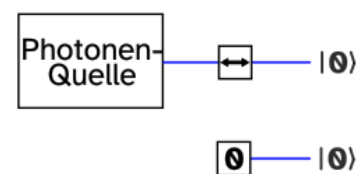


Filter und das Qubit geht durch, dann ist sein Zustand eben  $|1\rangle$ . Nach der Kopenhagener Interpretation kommt es nach der Messung zum Zusammenbruch der Wellen-Funktion.

Man spricht auch von Projektion. Es wird ein diverses Qubit entweder auf den Basis-Zustand  $|0\rangle$  oder  $|1\rangle$  geprüft.

Verwenden lässt sich dies, um Qubit's in einem definierten Zustand zu produzieren.

Setzt man eine Qubit-Quelle (Photonen-Quelle) vor ein Polarisations-Filter, dann erhält man hinter dem Gatter (Gitter) nur Qubit eines Basis-Zustand's.



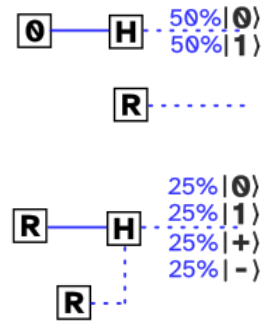
Als Vereinfachung können wir dann gleich so eine Art 0-Gatter schreiben. Dieses produziert nur den Basis-Zustand  $|0\rangle$ . Man spricht auch von einem **(Qubit-)Generator**.

Äquivalent kann auch ein 1-Gatter erstellt werden, welches eben nur Qubit's mit dem Basis-Zustand  $|1\rangle$  erzeugt.

Für viele Zwecke benötigt man einen **Zufalls-Generator**. Dieser soll die Basis-Zustände  $|0\rangle$  und  $|1\rangle$  mit exakt 50%iger Wahrscheinlichkeit erzeugen.

Ein solcher Generator lässt sich aus einem 0-Generator und einem nachgeschalteten HADAMARD-Gatter ( $\rightarrow$  [HADAMARD-Gatter, H-Gatter](#)) erstellen. Auch hier kann ein vereinfachtes Symbol für den Zufalls-Generator eingeführt werden. Das R steht dabei für Random bzw. verkürzt Rand.

Mit einem solchen Zufalls-Generator für die beiden Basis-Zustände lässt sich nun wiederum ein Zufalls-Generator für die Zustände  $|0\rangle$ ,  $|1\rangle$ ,  $|+\rangle$  und  $|-\rangle$  erstellen. Dazu wird hinter dem Zufalls-Generator noch ein weiteres HADAMARD-Gatter positioniert, welches aber zufällig aktiviert wird. Es handelt sich quasi um ein controlled-H-Gatter. So erhält man dann die Zustände  $|0\rangle$ ,  $|1\rangle$ ,  $|+\rangle$  und  $|-\rangle$ .





### 1.4.4. Quanten-Gatter zu Quanten-Schaltkreisen kombiniert

Wiederholung von weiter vorne:

Quanten-Schaltkreis					
	Eingang		nach Gatter	Messung	Ausgang (Wahrscheinlichkeit)
Register-Zustand	$ 0\rangle$	$1 *  0\rangle + 0 *  1\rangle$	$\frac{1}{\sqrt{2}} *  0\rangle + \frac{1}{\sqrt{2}} *  1\rangle$	od. $ 0\rangle$ $ 1\rangle$	50 % bzw. 0,5 50 % bzw. 0,5

betrachtet nun für Quanten-Register mit zwei Qubit's

#### Superposition mit Verschränkung

Superposition bedeutet, dass eben mehrere Ergebnisse möglich sind (mit ausreichend großen Wahrscheinlichkeiten)

Verschränkung bedeutet, dass zwei Qubit's miteinander verbunden (über die imaginäre Fern-Wirkung von Quanten)

Ausgangs-Situation ist  $|00\rangle$   
wohl-definiert (also eindeutig)  
Analyse im Zustands-Raum

Quanten-Schaltkreis						
	Eingang		nach 1. Gatter		Ausgang	
Register-Zustände	$ 0\rangle$	$1 *  0\rangle + 0 *  1\rangle$	$\frac{1}{\sqrt{2}} *  00\rangle + \frac{1}{\sqrt{2}} *  01\rangle$	CNOT <sub>1,2</sub>	$\frac{1}{\sqrt{2}} *  00\rangle + 0 *  01\rangle$	50 %
	$ 0\rangle$	$1 *  0\rangle + 0 *  1\rangle$	$1 *  00\rangle + 0 *  01\rangle$		$0 *  10\rangle + \frac{1}{\sqrt{2}} *  11\rangle$	50 %
Visualisierung						

## Superposition ohne Verschränkung

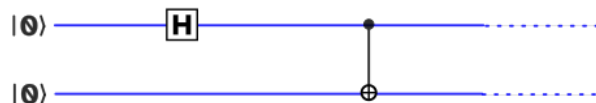
Quanten-Schalt-kreis			
	<b>Eingang</b>	nach 1. Gatter	<b>Ausgang</b>
Register-Zustände	$ 0\rangle$ $1 *  0\rangle$ $+ 0 *  1\rangle$	$\xrightarrow{H}$	$\frac{1}{\sqrt{2}} *  00\rangle$ $+ \frac{1}{\sqrt{2}} *  01\rangle$
	$ 0\rangle$ $1 *  0\rangle$ $+ 0 *  1\rangle$	$\xrightarrow{H}$	$\frac{1}{2} *  00\rangle$ $+ \frac{1}{2} *  01\rangle$ $+ \frac{1}{2} *  10\rangle$ $+ \frac{1}{2} *  11\rangle$
Visualisierung			

## Kombination eines HADAMARD- und eines CNOT-Gatters zur Erzeugung eines verschränkten System's

Schaltkreis liefert aus einem  $|00\rangle$  als Eingang den verschränkten Ausgang  $|\Psi\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$

schrittweise:

$$|00\rangle \Rightarrow \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \Rightarrow \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$



## Cloning-Maschine

dieses Gatter / dieser Schaltkreis soll ein beliebiges Qubit  $|q\rangle = \alpha|0\rangle + \beta|1\rangle$  oder einen der Basis-Zustände  $|0\rangle$  bzw.  $|1\rangle$  kopieren  
im Black-Box-Modell würde das so aussehen



die Cloning-Maschine müsste also

$$\begin{aligned}
 U(\alpha|0\rangle + \beta|1\rangle)|?\rangle &= (\alpha|0\rangle + \beta|1\rangle) (\alpha|0\rangle + \beta|1\rangle) \\
 &= \alpha^2|0\rangle|0\rangle + \alpha\beta|0\rangle|1\rangle + \alpha\beta|1\rangle|0\rangle + \beta^2|1\rangle|1\rangle
 \end{aligned}$$

erfüllen. Dabei stoßen wir auf ein Problem. Verwendet man statt der obigen Gleichung eine anderes Ausmultiplikations-Verfahren, dann ergibt sich

---

$$U(\alpha|0\rangle|?\rangle + \beta|1\rangle|?\rangle) = \alpha U|0\rangle|?\rangle + \beta U|1\rangle|?\rangle = \alpha|0\rangle|0\rangle + \beta|1\rangle|1\rangle$$

was eben ein Widerspruch zur obigen Gleichung darstellt. Daraus folgt, dass eine Cloning-Maschine nicht existieren kann. Man spricht auch vom **No-Cloning-Theorem**.

---

## 1.4.5. Quanten-Algorithmen

### SHOR-Algorithmus (1984)

faktoriert eine Zahl in polynominaler Zeit (bei "normalen" Rechner ist die Komplexität  $O(?)$ )  
benutzt FOURIER-Transformation; zerlegt eine Funktion in Frequenzen  
exponentiell schneller als normaler Computer  
große Bedeutung für die Kryptographie, da die Zerlegung einer Zahl in Primzahl-Faktoren, wegen des nicht-trivialen Bestimmen's von Primzahlen einfaches Ermitteln der Schlüssel verhindert

sind sehr schnell

### BB84-Protokoll (1984)

Verfahren zum sicheren Austausch von privaten Schlüsseln für eine sichere Kommunikation

### GROVER (+2)

schneller Such-Algorithmus für große, unsortierte Datenbanken  
man beobachtet quadratische Beschleunigung  
Komplexität hier  $O(\sqrt{N})$  sonst  $O(?)$

Simulationen im Bereich Biochemie, Chemie und Medizin, Stau-Vermeidungen

### DEUTSCH-JOSZA-Algorithmus

ermittelt ob eine Funktion balanciert ( $f(0) \neq f(1)$ ) oder konstant ( $f(0) = f(1)$ ) ist  
löst das Problem von DEUTSCH  
Quanten-Computer lösen das Problem mit nur einem Zugriff auf die Funktion

viele (echte) Probleme (in der klassischen "Bit"-Informatik) haben die Komplexität  $O(2^n)$ ,  
bei Quanten-Algorithmen sinkt / ist die Komplexität nur noch  $O(n)$

### abhörsichere Kommunikation

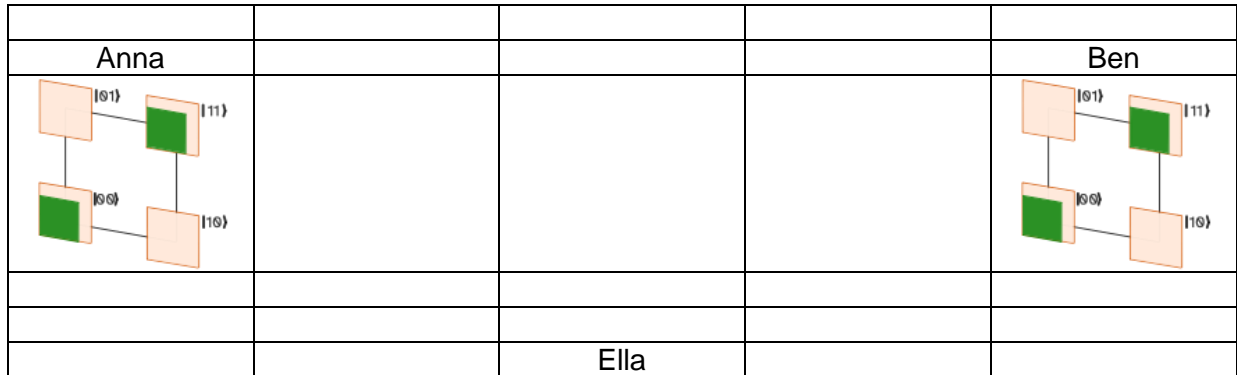
wenn Anna und Ben sich unterhalten und eine geheime Nachricht austauschen wollen, dann könnte eine Zwischen-Person (hier Ella) die Nachricht mithören (Man-in-Middle-Angriff) oder sich gar mal als Ben und mal als Anna ausgeben und die Nachricht(en) abfangen und manipulieren

ein sicherer Austausch von Schlüsseln für eine klassisch-verschlüsselte Kommunikation ist so nicht möglich, Ella könnte sich schon in die erste Kommunikation bzw. die Kommunikations-Anbahnung einmischen

bei entfernter Kommunikation, und besonders, wenn sich die Kommunikanten nicht kennen, dann ist eine sichere Kommunikation kaum möglich

mit Quanten-Systemen ist eine sichere Kommunikation und vor allem ein sicherer Schlüssel-Tausch für eine klassisch-verschlüsselte Kommunikation möglich  
beide Kommunikanten haben ein verschränktes Quanten-Register

jetzt misst einer der beiden Kommunikanten sein Quanten-Register  
in unserem Beispiel kann jetzt nur  $|00\rangle$  oder  $|11\rangle$  herauskommen  
der messende Kommunikant teilt dem anderen nun mit "Ich habe gemessen!" ohne auch  
etwas über das mess-Ergebnis selbst mitzuteilen  
nun kann der andere Kommunikat sein Quanten-System abfragen und erhält, wegen der  
Verschränkung, das gleiche Ergebnis und kann ebenfalls mitteilen, das er gemessen hat  
auch hier wird das Ergebnis nicht mitgeteilt  
der Mithörer bekommt also nur die Information, dass ein Schlüssel od.ä. ausgetauscht / er-  
stellt wurde, aber garnichts über dessen Inhalt



bei ausreichend vielen verschränkten Quanten-Bit's kann das Ergebnis auch nicht mehr  
sinnvoll vorausgesagt / erraten werden  
hier – bei 2 Qubit's würde die Chance beim Erraten bei 50% liegen

Problem ist der Austausch der verschränkten Quanten-Bit's  
wenn Ella sich da einmischt ist ein "Men-in-the-Middle"-Angriff weiterhin denkbar

*Idee: Modell der weichen Passung*

klassische Prüfung testet mögliches Ergeb-  
nis für Ergebnis den Algorithmus / das ge-  
stellte Problem

auch schrittweise Testung einzelner Zwi-  
schen-Lösungen möglich

der weichen (An-)Passung der möglichen  
Ergebnisse an einen Algorithmus  
alle möglichen Ergebnisse werden quasi  
gleichzeitig geprüft, wobei eben nur einzelne  
oder wenige Ergebnisse passen

1.4.6. Quanten-Computer

Bedingungen, die ein System erfüllen muss, um als Quanten-Computer genutzt zu werden:

---

## Bedingungen

- |                           |                                                                                                                                                |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1. Messbarkeit</b>     | der Ausgang (Output) des System's muss messbar / lesbar sein, d.h. Qubit für Qubit                                                             |
| <b>2. Universalität</b>   | es werden eine realisierbare Auswahl von Quanten-Gattern benötigt, mit deren Hilfe beliebige Operation ausgeführt werden können                |
| <b>3. Skalierbarkeit</b>  | das System muss um weitere Qubit's erweiterbar sein, ohne dass die Quanten-Zustände zerstört / verändert werden                                |
| <b>4. Initialisierung</b> | alle Qubit's müssen in den Zustand $ 0\rangle$ versetzbar sein                                                                                 |
| <b>5. Kohärenz</b>        | die einzelnen Quanten-Zustände müssen stabil gegen Dekohärenz sein, so dass man Fehler-Strategien in einem umsetzbaren Rahmen realisieren kann |

### Esels-Brücke MUSIK

Problem ist die notwendige Isolation der Qubit's von der Umgebung, damit keine Dekohärenz eintritt (und damit die Qubit's ihre Superpositions-Eigenschaft verlieren)  
gleichzeit müssen die Qubit's aber auch manipulierbar sein, damit Eingaben und Initialisierungen gesetzt werden können  
Ziel ist ein bestmöglicher Kompromiß

unbewegliche Qubit's interagieren nur mit ihren unmittelbaren Nachbarn (eine Langstrecken-Kommunikation (Verschrängung mit entfernten Qubit's) ist nur über spezielle Swap-Gate's möglich  
bei beweglichen Qubit's ist es möglich, dass alle Qubit's miteinander kommunizieren können

## Definition(en): Topologie

Die Topologie ist die Beschreibung der Verbindbarkeit / Kombinierbarkeit der einzelnen Qubit's, um eine Verschrängung zu erzielen.

---

## 1.5. Virtualisierung

Haupt-Zielrichtung  
Unabhängigkeit von der Hardware  
Erhöhung der Sicherheit  
Vereinheitlichung / Standardisierung der Systeme

### Virtualisierung / Simulation von Rechnern / Servern

#### **Vorteile:**

- Einsparung an Hardware
- Einsparung an Energie (praktische läuft nur noch ein echter Rechner)
- Mehr-Verbrauch an Energie nur, wenn die Client's wirklich arbeiten
- effektivere Nutzung der Ressourcen
- meist bessere Verfügbarkeit
- einfaches Erstellen von vielen gleichartigen Servern
- schnelleres Auf- und Runter-Skalieren (Server-Anzahl)
- gute Isolation von Servern
- isoliertes Testen von Software
- Bereitstellung von Honey-Pot's (für Firewall-Systeme) → Beobachtung des Hacker-Verhalten's; Bereitstellung von irrelevanten Daten in großer Menge
- gute Wiederherstellbarkeit bei Ausfällen
- ...

#### **Nachteile:**

- Host-Rechner muss Leistungs-stärker sein; braucht mehr Speicher (ev. bessere CPU)
- Ausfall / Fehler des Host's beeinflusst mehrere Client's
- bei gleichzeitiger Hoch-Belastung der Client's kommt der Host schnell an seine Grenzen
- ...

Komponente	Energie-Verbrauch [W]		
	Leerlauf	Voll-Last	
Computer (PC)	120	300	
Bildschirm	3	40	
Laptop	30	75	
Drucker (Laser-)	5	400	
Drucker (Tintenstrahl-)	3	200	
(kleiner) Server	150	400	
(großer) Server mit virtualisierten Servern	350	750	
Smartphone			

---

### Aufgaben:

1. *Vergleichen Sie quantitativ den Leerlauf-Energie-Verbrauch von 3 kleinen (vollständigen) Servern mit einem großen (vollständigen) Server! Der große Server soll pro virtualisiertem Server ungefähr 10% mehr Energie verbrauchen.*
2. *Ermitteln Sie mit Hilfe eines Messgerätes für den Strom-Verbrauch den täglichen Energie-Bedarf Ihres Arbeits-Rechner's (PC, Tablet, Laptop, ...) und Ihres Smartphone's (Laden)!*
3. *Berechnen Sie die Tages- und Jahres-Energie-Kosten zu den obigen Aufgaben für einen Strom-Preis von 45 Cent pro kWh!*

### Einbau einer Zwischen-Schicht

bietet den Programmier-Systemen eine einheitliche Hardware Unterbindung des direkten Zugriff's auf Hardware-Komponenten, damit Einschränkung der (feindlichen) Hardware-Manipulationen und des Ressourcen-Verbrauch's meist Kontrolle der Zugriffe durch die Zwischen-Schicht

### Emulatoren

bildet ausgewählte Funktionen eines anderen System's nach  
z.B. um Atari-Spiele auf einem PC nutzen zu können

z.B. Snes9x emuliert ein Super-Nintendo-System auf dem PC  
benötigt werden nur die sonst im ROM gespeicherten Spiel-Programme als Datei  
z.B. verfügbar auf: <https://emulatorgames.net/roms/super-nintendo/>

### Java Runtime Environment (JRE)

vereinheitlicht Zugriffs-Möglichkeiten der JAVA-Programme auf die lokale Hardware und das laufende Betriebssystem

### .Net

### Wine

erzeugt in einem Linux-System eine Schnittstelle zu (nachgebauten) Windows-Funktionen



---

## echte Virtualisierung von Rechner-Systemen

Host ist lokaler Rechner

### *virtualBox*

<https://virtualbox.org>

praktisch frei verfügbar

natürlich ev. Lizenzen für die gehosteten Systeme notwendig

### *VMware*

### *Docker*

## Cloud-Computing

Bereitstellung von Programmen (Apps) oder Systemen über das Internet oder das eigene Netz

Host ist ein Internet-Server

<https://owncloud.org>

bietet diverse Nutz-Leistungen

- für eigene Rechner nutzbar (z.B. auch auf einem Raspberry Pi)  
viele Cloud-Anbieter nutzen owncloud für ihr Angebot

auch eine virtuelle Maschine für virtualBox vorhanden  
(eigener Server im (virtuellem) Netz)

### ***Typen des Cloud-Computing***

- **Infrastructure as a Service**

**IaaS**

Online-Bereitstellung von Computer-Hardware, z.B. virtuelle Maschinen, Speicher, Netzwerk-Komponenten

- **Platform as aService**

**PaaS**

---

Bereitstellung einer Cloud-basierten Umgebung für das Programmieren und Bereitstellen von App's  
z.B. Systeme der Künstlichen Intelligenz

- **Software as a Service**

**SaaS**

Online-Bereitstellung von App's und Datenbanken  
z.B. Nutzung von Office-Programmen oder Warenwirtschaftssystemen (/ von Rechen-Leistung) auf dem Systemen des Anbieter's (im Abonnement)

---

## 1.6. Kenndaten für Computer-Systeme

$$\text{Verfügbarkeit} = \frac{\text{Gesamtlaufzeit} - \text{Gesamtausfallzeit}}{\text{Gesamtlaufzeit}}$$

### Aufgaben:

1. Die nachfolgenden Server sind im letzten Jahr

A: 2 d                      B: 16,4 h                      C: 348 min                      D: 55 min

ausgefallen. Berechnen Sie die Verfügbarkeit der Server!

2.

3. Berechnen Sie die mögliche Ausfallzeit (in Stunden bzw. Minuten) für ein System bei dem

a) 99 %                      b) 99,5 %                      c) 99,9 %                      d) 99,99 %

garantiert worden sind!

---

## 2. Netzwerke - Grundlagen

### 2.1. Grundlagen Netzwerke

#### **Definition(en): Netzwerk**

Ein Netzwerk (im informatischen Sinne) ist eine Daten-austauschen Verbindung von mindestens zwei – meist aber viel mehr – Datenverarbeitungs-Geräten.


oft synonym verwendet:

Netz, Datennetz, Net, PC-Netz, LAN, Computernetzwerk

Node

ist ein Gerät, das über eine od. mehrere Schnittstellen (Interfaces) mit eine, od. mehreren Netzwerken verbunden ist.

Host

ist eine Node ohne Router-Eigenschaften  
stellt ein Endgerät dar (meist auch nur eine Schnittstelle)  
allgemein werden Server und Clients als Hosts bezeichnet

Knoten

Verzweigungspunkt eines (Kommunikations- / Datenübertragungs-)Netzwerkes  
z.B. Vermittlungsstellen, Router, ...  
sleten werden auch Zugangspunkte zu Netzwerken als Knoten bezeichnet

---

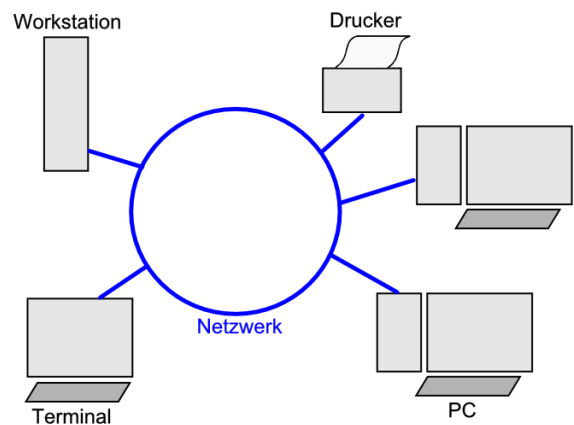
## Topologien

### **Topologie: Struktur-Aspekt**

Anordnung und Verbindung der Netzgeräte untereinander

#### **Ring-Topologie**

jede Station hat zwei Nachbarn  
wegen der technischen Einfachheit im Um-  
satz wurde diese Topologie früher sehr häu-  
fig gewählt

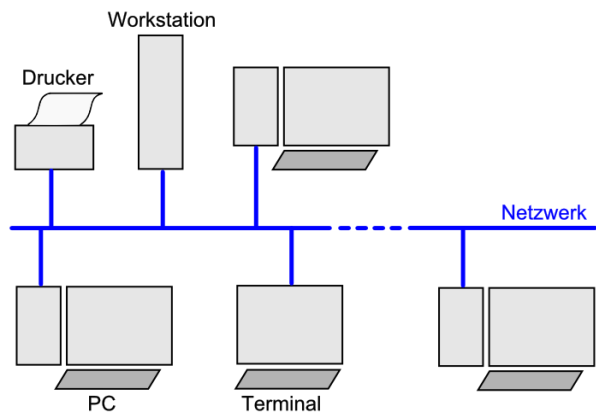


Vorteile:  
geschlossene, sichere Struktur (Manipulation ist bemerkbar)

Nachteile:  
für entfernte Stationen doppelte (lange) Verkabelung notwendig  
spezielle Einkoppel-Interfaces notwendig

## Bus-Topologie

alle Geräte benutzen und teilen sich ein zentrales Kabel / Medium



Vorteile:

geringe Kosten

einzelne Station beeinflusst bei eigenen Störungen / Defekten die anderen Station nur geringfügig

Nachteile:

zentrales Kabel / Medium bestimmt Qualität und Quantität des Datenverkehrs

Unterbrechung des zentralen Mediums verhindert gesamte Kommunikation

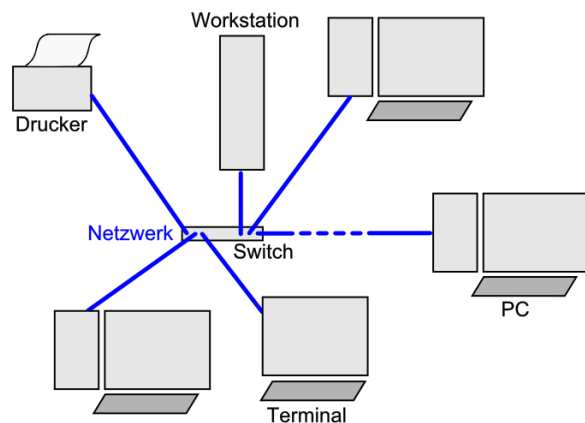
unverschlüsselter Datenverkehr kann von jeder Station mitgelesen werden

immer nur eine Station kann senden

## Stern-Topologie

jedes Gerät hat ein eigenes Kabel / Medium

die Kabel / Medien werden über eine zentrale Einheit (Hub od. Switch) kommunikativ miteinander verbunden



Vorteile:

Störung an einer Station oder seinem Kabel beeinflusst kaum die Kommunikation der anderen Stationen

zentrale Station (Hub, Switch) ist gleichzeitig Signal-Verstärker

problemloses Erweitern und Abbauen des Netzes (quasi im laufenden Betrieb möglich)

---

Schachtelung möglich

Nachteile:

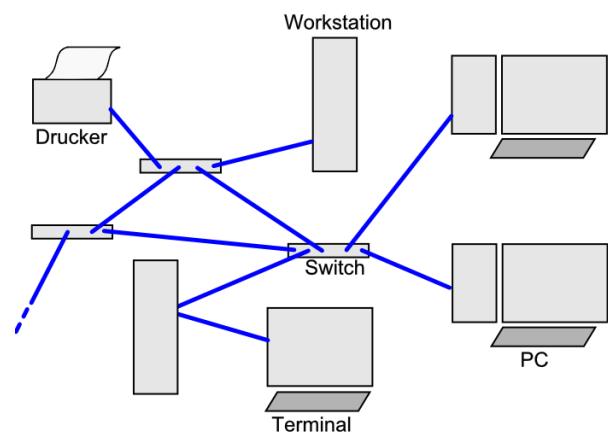
zentrale Einheit bestimmt Anzahl der Verbindungen und Datendurchsatz

zentrale Station (Hub, Switch) kostet zusätzlich

zentrale Station ist kritisches Element der Struktur (bei Ausfall versagt das gesamte Netz)  
(kein einfaches Reparieren möglich)

### **Maschen-Netz als offene Misch-Topologie**

Kombination aus anderen Topologien,  
meist unter Bildung von Maschen



Vorteile:

hohe Redundanz / Ausfallsicherheit

Lasten-Verteilung möglich

Nachteile:

unübersichtlich

schwer kontrollierbar

---

Simulationen mit Snap!

s.a. Kurs OpenSAP MODROW: Einführung Informatik (2. Wo. 6. Einheit "Konnektivität")

Netzwerke / Graphen

interessante Probleme:

Existiert ein zusammenhängender Graph für alle Knoten? (Kann von jedem Knoten zu jedem anderen Knoten kommen kann?)

Welche kürzeste Wege existieren zwischen zwei Knoten (→ Routen-Planung)?

Wieviele Verbindungen sind minimal notwendig um eine bestimmte Anzahl von Knoten (vollständig) zu verbinden?

Wieviele Anzahl von Verbindungen sind (durchschnittlich) notwendig, um von einem Knoten zu einem (beliebigen) anderen Knoten zu gelangen?

Welche Kanten / Verbindungen müssen zerstört werden, um einen Graphen zu zerlegen (damit er nicht mehr vollständig ist)?

Anfälligkeit von Netzen gegenüber zufälligen Schäden?

Anfälligkeit von Netzen gegenüber gezielten Schäden (z.B. Hub's ausschalten / angreifen)?



---

## **weitere Topologien**

### **Linie**

Geräte sind hintereinander angeordnet (z.B. Signal-Verstärker auf einer Leitung)

### **Baum**

Geräte sind hierarchisch / Baum-artig angeordnet; bis auf das Wurzel-Gerät hat jedes gerät ein übergeordnetes Gerät und kann ein bis zwei (Baum-artig) bzw. noch (hierarchisch) mehr untergeordnete Geräte haben / steuern  
teilweise in Telekommunikations- / Funktelefon-Netzen realisiert

### **Vollverbindung**

jedes Netzgerät ist mit jedem weiteren Netzgerät verbunden  
jedes Gerät muss soviele Ports besitzen, wie es weitere Geräte im Netz gibt

bei Hochleistungsrechner / Hochleistungs-System (z.B. bestimmte Multi-Kern-CPU's)

---

## Topologie: Ausdehnungs-Aspekt

räumliche Ausdehnung / Dimension des Netzes

### GAN

Global Area Network  
Kontinente, gesamte Erde (Internet)

Definition(en): GAN

### WAN

Wide Area Network  
Land, Länder-Verbund

Definition(en): WAN

---

## MAN

LANs und MANs

→ KALDEALI → S. 66 ff.

<b>Definition(en): MAN – Metro Area Network</b>
Ein MAN (sprich: mahn) ist ein regionales Netzwerk, das sich i.A. über mehrere Grundstücke / Gebäude / Werkhallen / eine große Firma erstreckt.

---

## LAN

### **Definition(en): LAN – Local Area Network**

Ein LAN (sprich: lahn) ist ein räumlich auf den Ort / eine größeres Grundstück / eine kleine Firma beschränktes Netzwerk.

Im Allgemeinen werden die Grenzen bei rund 500 m Ausdehnung gezogen.

## PAN

begrifflich weniger genutzt

i.A. als LAN geführt

praktisch gleiche Technik

### **Definition(en): PAN – Personal Area Network**

Ein PAN (sprich: pahn) ist ein räumlich auf den Standort / das Grundstück / das Haus / die Wohnung beschränktes Netzwerk.

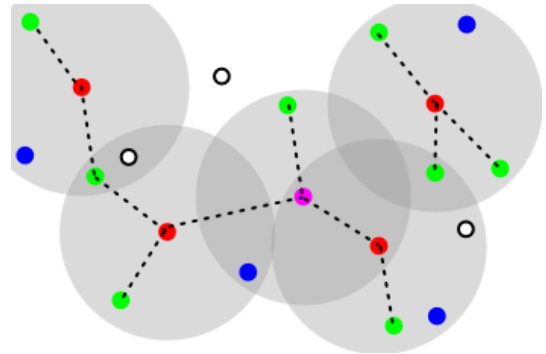
Im Allgemeinen werden die Grenzen bei rund 500 m Ausdehnung gezogen.

---

## Ad-hoc- und Pico-Netzwerke

dynamische Netz-Struktur  
jedes Gerät kann jede Funktion (Sender, Repeater, Empfänger) übernehmen

bei Verbindungs-Abbruch (Bewegung, Ausschaltung) werden neue Verbindungen aufgebaut



Scatternet / Ad-hoc-Netz(e)  
(Master: rot; Slave: grün; Master und Slave: violett; geparkt: blau; ausgeschaltet: schwarz/weiß)

## Bluetooth



offizielles Logo  
Q: de.wikipedia.org

---

## **RFID – Radio Frequency Identification**

Reichweite bis mehrere Meter  
meist zwischen 0,5 und 2 m

passive RFID-Transponder

aktive RFID-Transponder

## **NFC – Near Field Communication**



Beispiel-Logo's  
für NFC

---

## **Topologie: (inhaltlicher) Abgrenzungs-Aspekt**

Größe des Nutzerkreises und Verfügbarkeit der Informationen

### **Intranet**

nicht öffentlich; bestimmtes (geschütztes) Netzwerk-Segment  
Institutions-intern  
eingeschränkte Funktionen / Protokolle / Ressourcen  
beschränkt dynamisch (geplante Dynamik)  
verlässlich  
Rechte-basierte Nutzung (Login notwendig für Inhalts-Zugriff)

Bereitstellung von (Institutions-)internen Datenbanken, Dokumenten, Dateien, Programmen  
Kommunikation zwischen Mitarbeitern, Abteilungen, Hierarchie-Ebenen  
Organisation von Abläufen / Prozessen  
Datenschutz / Datensicherheit verbessern / realisieren

<b>Definition(en): Intranet</b>
Ein Intranet ist ein beschränktes Netzwerk-Segment eines Besitzers, in dem die eigenen Daten und deren Nutzung im Vordergrund stehen.

---

## **Internet**

verkürzt aus internetwork  
heute häufig nur noch das Netz genannt

sachlich ist das Netz zwischen den Netzen – das Zwischennetz, Weltnetz - gemeint

öffentlich; nur interne Netzwerke an den Rändern des Internet sind eingeschränkt  
offen strukturiert, dynamisch (zufällig, nicht planbar)  
unzuverlässig, viele Alternativen vorhanden (Netz-Neutralität)  
Login notwendig (für Abrechnung)

### **Definition(en): Internet**

Das Internet ist die Netzwerk-Struktur, die verschiedene andere Netzwerk-Segmente zu einem Gesamtgebilde verknüpft und die gemeinsame, freie Nutzung der Daten zum Ziel hat.

Das Internet ist die übergreifende, weltweite Verbindung von autonomen Netzwerk-Segmenten.



---

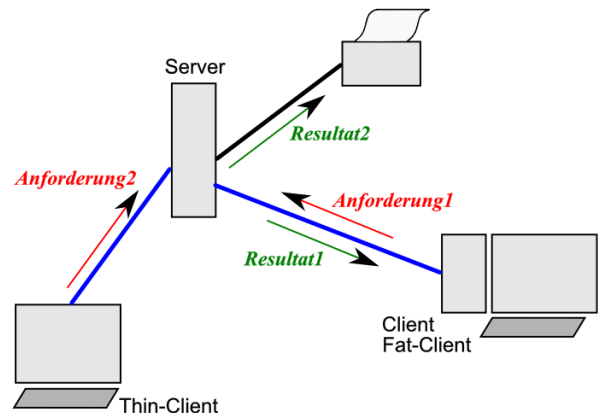
## Kommunikations-Konzepte

Client-Server-Paradigmen, ...

→ KALDEALI → S. 367 ff.

### Client-Server-Konzept

ein Gerät stellt bestimmte Leistungen im Netz zur Verfügung  
z.B. einen Drucker, Verzeichnisse mit Dateien, eine Datenbank, ...  
ein oder mehrere Client's stellen Anforderungen an den Server  
dieser verarbeitet diese und schickt das Ergebnis / eine Antwort zurück bzw. erfüllt die Leistungs-Anforderung (z.B. Ausdruck, Versand von eMails, ...)



es reicht oft ein leistungsstärkerer Server für viele – ohne weiteres auch etwas schwächere – Client's

in modernen Systemen werden die sonst üblichen Fat-Clients (meist vollständige PC's) gegen minimal ausgestattete Thin-Clients ausgetauscht

in dem Fall übernimmt der Server fast alle Aufgaben des Clients, nur noch Ein- und Ausgabe wird am Thin-Client (Terminal) realisiert

Thin-Clients haben keine Festplatte mehr; Betriebssystem wird über Netzwerk in den Speicher geladen und weitgehend hier verarbeitet, alle weiteren Leistungen müssen immer über das Netzwerk angefordert und entweder in den Client-Speicher übertragen werden oder der Server muss die Aufgabe realisieren

man spricht auch von Terminal-Server-Lösungen; sehr leistungsstarke und gut ausgestattete Server notwendig

geringer Arbeitsaufwand für Administratoren

Anzahl der Server typischerweise kleiner als die der Clients

z.B.:

www, ftp, eMail,

## Peer-to-Peer-Konzept

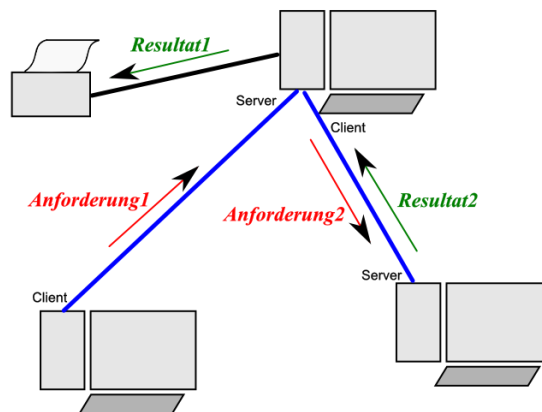
Stationen sind – bis auf Peripherie –  
gleichberechtigt im Netz  
Jedes Gerät kann sowohl als Client, als  
auch als Server im Netzwerk arbeiten

meist werden nur einfache Leistungen  
von den einzelnen Stationen bereitge-  
stellt

zu viele gleichzeitige Anforderungen  
führen schnell zur Überlastung einer  
Station

Rechte-Systeme nur schwer durchsetz-  
bar

z.B.:  
Filesharing / Tauschbörsen, VCoIP ( over IP)



Unicast  
es besteht eine 1 : 1-Verbindung zwischen zwei Stationen (direkte od. indirekte Verbindung  
(über ein Netzwerk))  
unidirektional od. bidirektional

Broadcast  
ist eine 1 : n-Verbindung, ein Sender spricht alle Empfänger / anderen Netz-Teilnehmer an  
z.B. Rundfunk, Fernsehen

Multicast  
ein Sender spricht alle Geräte / Empfänger einer Gruppe an  
erst der letzte Router verteilt die Daten auf die einzelnen Endpunkt-Leitungen  
dadurch rel. geringer Verbrauch an Bandbreite (bis zum letzten Router werden die Daten nur  
einmal übertragen)

Anycast  
es besteht eine Verbindung zwischen einem Sender und einem Empfänger aus einer Gruppe  
von Empfängern (welches Gerät reagiert / empfängt ist egal / unterliegt dem Zufall)  
zur Kommunikation mit Geräten, die alle die gleiche Adresse (also eine Anycast-Adresse)  
besitzen  
keine Verbindungs-orientierte Datenübertragung möglich, da nicht sicher ist, welches Gerät  
angprochen worden ist

---

## 2.2. Grundlagen Datenübertragung

### 2.2.x. allgemeines Modell der Kommunikation

Warren WEAVER + Claude E. SHANNON (1949)  
mathematische Theorie der Kommunikation / Informations-Theorie



Quelle → Kodierer → Übertragungskanal → Decodierer → Senke

Quelle → Signalaufbereitung → Übertragungskanal → Signlrückgewinnung → Senke

Quelle → Modulator → Übertragungskanal → Demodulator → Senke

zu übertragende Information wird über den Kodierer in ein übertragbares (für den Übertragungsknale geeignetes) Signal umgesetzt

der Dekodierer wandelt das übertragene Signal wieder in eine nutzbare Information um

besonders auf den Übertragungskanal wirken Störungen

Gegenmaßnahmen sind Schutz des Kanals (Isolation, Abschirmung, ...) und redundante Signale

einseitige und wechselseitige Kommunikation

---

## 2.2.x. Medien für die Daten-Übertragung

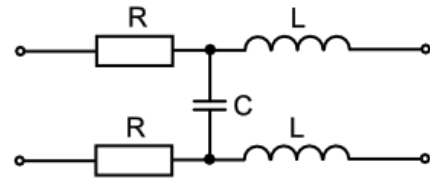
### 2.2.x.y. Kabel

elektrisches Feld um Leiter  
im Gleichstrom

bei entgegengesetzten Stromflüssen in parallelen Leitern (z.B. eines Kabels) treten schon erste Beeinflussungen / Störungen auf

da über die Leitungen aber viele An-Aus-Signale (0 oder 1) transportiert werden müssen, haben wir es praktisch mit einem Wechselstrom zu tun  
daraus folgen stärkere gegenseitige Beeinflussungen der Leitungen

ein Schaltbild-Modell für ein Kabel enthält außer den Leitungs-Widerständen nun auch Kapazitäten und Induktivitäten



Ersatzschaltung für eine Wechselstrom-Leitung

durch Abschirmungen (Ummantelung oder (Schirm-)Geflecht) können Störungen / beeinflussungen schon reduziert werden

insgesamt sind Leitungen aber immer beschränkt hinsichtlich der für eine Übertragungsfrequenz noch nutzbare Kabel-Länge

Twisted-Pair-Kabel – Kabel mit verdrehten Aderpaaren

vor allem bei vielen parallelen Leitungen

symmetrische Signal-Übertragung auf mehreren Kanälen

deshalb erstmals in der Computertechnik bei Centronics-Kabel (sehr alte parallele (Drucker-)Schnittstelle)

später dann auch bei SCSI-Verkabelungen von Festplatten, CD- bzw. DVD-Laufwerken und Motherboards

geschirmt → Shielded Twisted Pair → STP

ungeschirmt → Unshielded Twisted Pair → UTP

#### **interessante Links:**

<http://people.ee.ethz.ch/~pascal/Hochspann/> (Animation von elektrischen und magnetischen Feldern an Hochspannungsleitungen)

### 2.2.x.y.z. Luftpipeline

---

**2.2.x.y.z. Verlegekabel**

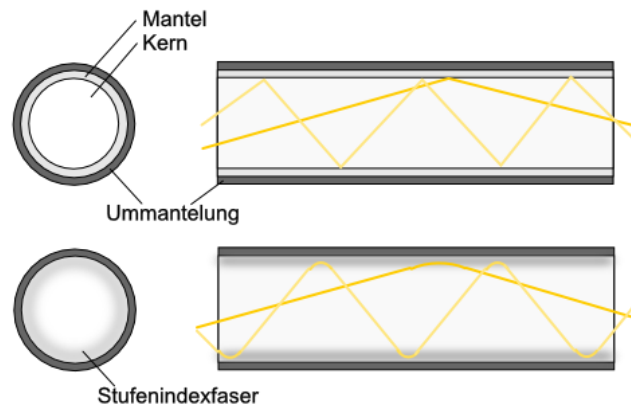
**2.2.x.y.z. Stromleitungen**

**2.2.x.y.z. Erdkabel**

**2.2.x.y.z. Unterseekabel**

---

## 2.2.x.y. Lichtwellenleiter



### 2.2.x.y.z. Single-Mode

üblicherweise UV-Licht  
zur Aktivitäts-Kennzeichnung zusätzlich noch rotes LED-Licht ohne Daten-Übertragungsfunktion

### 2.2.x.y.z. Multi-Mode

mit mehreren verschiedenen Wellenlängen parallel betrieben

---

## **2.2.x.y. Funkwellen**

### **2.2.x.y.z. Infrarot**

praktisch Licht aus Infrarot-Bereich (Wärme-Strahlung)

kabellos, funkähnlich

nur direkte Verbindung möglich, keine Gegenstände im direkten Sender-Empfänger-Weg möglich

→ Fernsteuerungen

→ IrDA-Schnittstelle einiger etwas älterer PC's

### **2.2.x.y.z. Richtfunk**

### **2.2.x.y.z. Satellitenfunk**

### **2.2.x.y.z. Landfunk**

### **2.2.x.y.z. Lokalfunk**

WLAN

Bluetooth

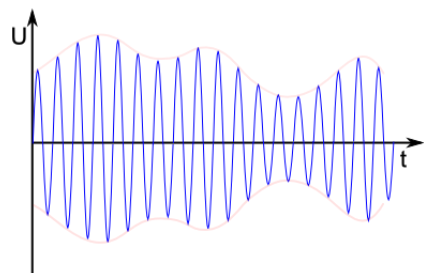
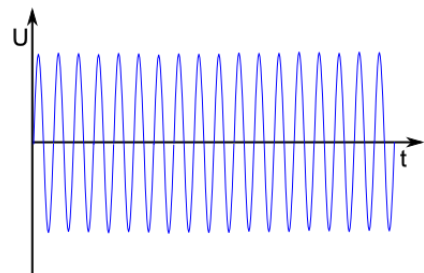
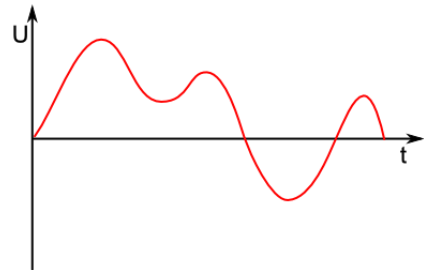
NFC

IrDA

---

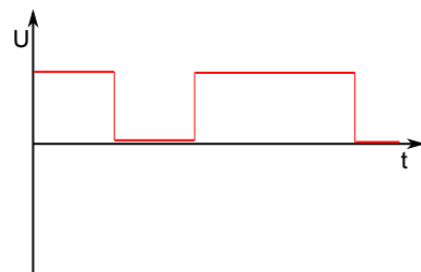
## 2.3.x. Modulations-Verfahren

### 2.3.x.y. Amplituden-Modulation



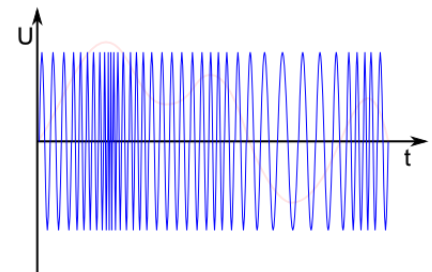
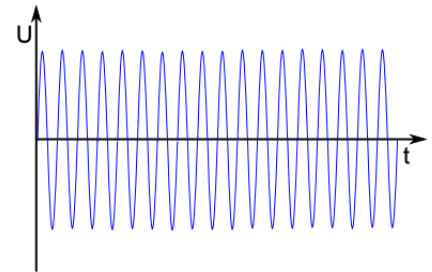
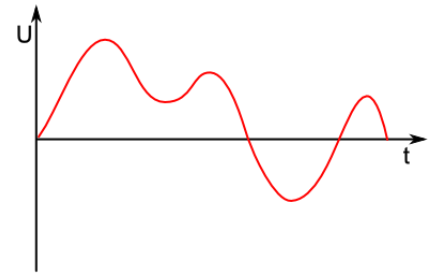
#### Aufgaben:

1. Erstellen Sie ein Modulations-Diagramm, wenn nebenstehendes Daten-Signal per Amplituden-Modulation codiert wird!
2. Beurteilen Sie, wie stör anfällig ein Amplituden-moduliertes Digital-Signal ist! Begründen Sie Ihre Meinung!



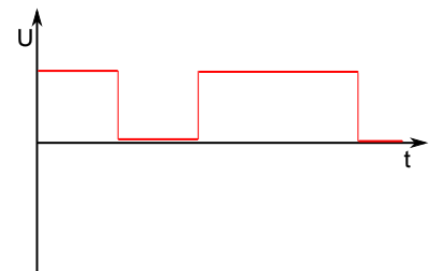


### 2.3.x.y. Frequenz-Modulation



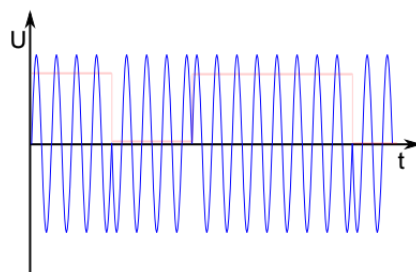
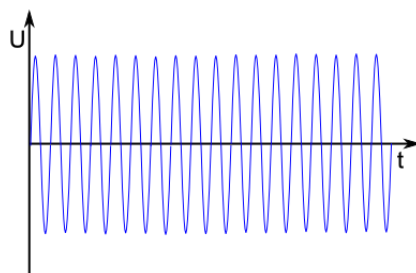
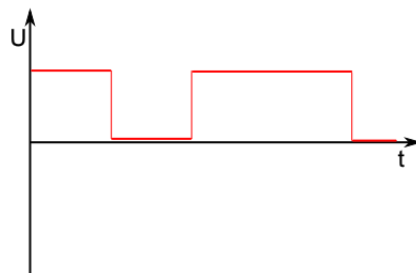
#### Aufgaben:

1. Erstellen Sie ein Modulations-Diagramm, wenn nebenstehendes Daten-Signal per Frequenz-Modulation codiert wird!
2. Beurteilen Sie, wie stör anfällig ein Frequenz-moduliertes Digital-Signal ist! Begründen Sie Ihre Meinung!



---

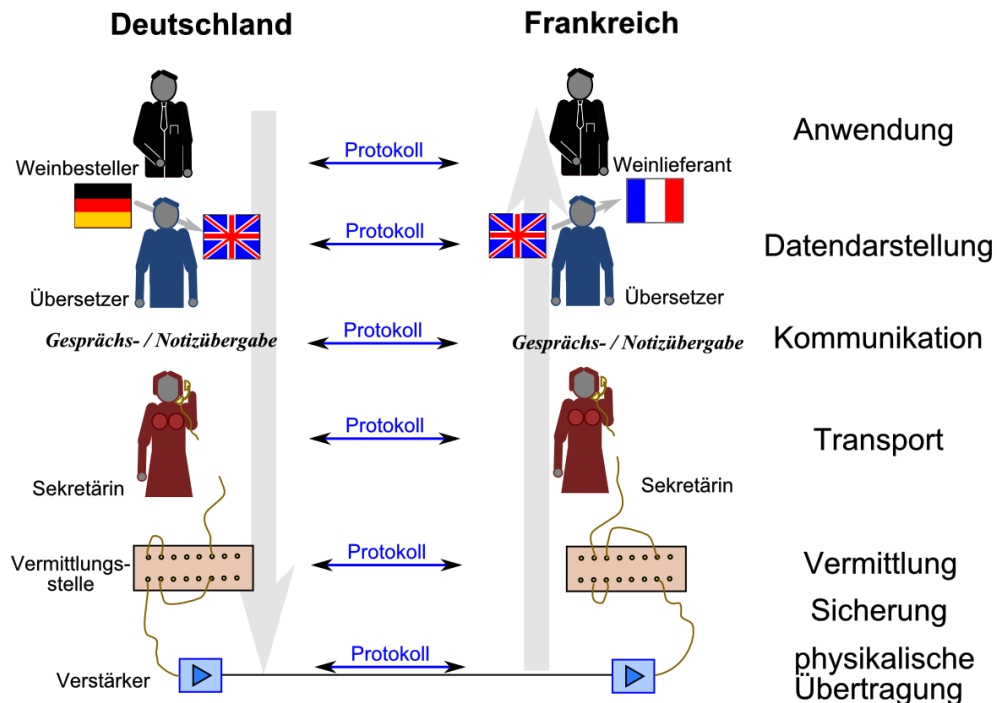
### 2.3.x.y. Phasen-Modulation



### Aufgaben:

## 2.4.x. Schicht-Modelle

### 2.4.x.y. ISO-OSI-Schichtmodell



Q: geändert aus /6/

ISO ... International Organization of Standardization  
OSI ... Open Systems Interconnection

Ziel war eine Aufgaben-spezifische Spezialisierung / Hierarchie -Struktur / Schichtung  
Schnittstellen-Definition; Definition abstrakter Aufgaben  
Geräte- und Anwendungs-spezifische Software-Komponenten

Spezialisierung auf jeder Ebene

Klärung von Verantwortlichkeiten, Erhöhung der Software-Qualität, effektive Aktualisierungen (Updates), Anwendung von Fachkenntnissen / Profiwissen auf bestimmten Ebenen (Vernetzung oder Betriebssystem usw.)

trotzdem ist Vereinigung von Schichten möglich und zulässig; Referenz-Modell; Empfehlungs- und Richtlinien-Charakter  
z.T. auch schon anderes realisiert, weil die Technologien z.T. älter als das Modell sind z.B. TCP

jede Schicht verfügt über Protokolle, so dass praktisch auf dieser Ebene kommuniziert werden kann  
in den meisten Fällen geht allerdings die praktische Verbindungs-Arbeit über die darunterliegenden Schichten (erst runter und dann wieder hoch)

---

auch: OSI-7-Layer-Model (OSI-7-Schichten-Modell)

ist ein Referenz-Modell; hat Empfehlungs-Charakter

jeder kann natürlich eine Software entwickeln, die direkt auf die Bit-Übertragung (Ethernet) zugreift oder gar ein eigenes Bit-Übertragungs-System (quasi ein Alternativ-Ethernet) entwickeln  
es fehlt dann aber Zusammenarbeit mit anderen Firmen und Programmen und breite Nutzung ist eingeschränkt

Schicht 7	Anwendungs-Schicht
Schicht 6	Darstellungs-Schicht
Schicht 5	Kommunikations-Steuerungs-Schicht
Schicht 4	Transport-Schicht
Schicht 3	Vermittlungs-Schicht
Schicht 2	Sicherungs-Schicht
Schicht 1	Bitübertragungs-Schicht

## Überblick über das OSI-Modell und Einordnung des TCP/IP-Modell's

OSI-Schicht-Modell				TCP/IP-Referenz-Modell				
OSI-Schicht	-Layer	-Schicht	Protokolle (Bsp.)	DoD Schicht	Einordnung	Protokolle	Einheit / Struktur-Größe	Kopplungs-Elemente
7	Application-	Anwendungs-		Anwendung	Ende-zu-Ende (Multihop)	TELNET, FTP, SMTP, DNS, HTTP, ...	Daten	Gateway, Content-Switch, Proxy Layer-4-7-Switch
6	Presentation-	Darstellungs-						
5	Session-	Kommunikations-Steuerungs-	NetBIOS	Transport				
4	Transport-	Transport-	X.224, SPX, NetBEUI			TCP, UDP	TCP → Segmente UDP → Datagramme	
3	Network-	Vermittlung-/Paket-	X.25	Internet	Punkt-zu-Punkt	IP, IPsec, IPX	Pakete	Router, Layer-3-Switch
2	Data-Link-	Sicherung	IEEE 802.11 (WLAN) IEEE 802.5 (Token-Ring)	Netzzugriff		ARPA, ALOHA, CSMA	Rahmen	Bridge, Layer-2-Switch
1	Physical	Bitübertragung	V.24, RS 423			Bit's, Symbole, Pakete	Netzwirkabel, Repeater, Hub	

**Definition(en): ISO-OSI-Modell**

Das ISO-OSI-Modell ist ein Referenz-Modell für eine Schicht-Architektur und Schicht-orientierten Netzwerk- bzw. Kommunikations-Protokollen.

Das ISO-OSI-Modell ist ein Vorschlag einer Schichtung von Kommunikations-Ebenen in Computernetzen, die für jede Schicht mögliche Verbindungs-Arten und –Protokolle vorschlägt.

TCP/IP- / DoD-Modell ist älter, historische Wurzeln liegen vor ISO-OSI-Modell, deshalb andere Schichtung, aus praktischen Gründen Zusammenlegung, intern (innerhalb der Software) oft aber aus technischen und programmiertechnischen Gründen Untergliederung nach ISO-OSI-Modell vorhanden (aber eben nicht notwendig); Spezialisierung und Hochtechnisierung nicht so weit fortgeschritten; Probleme waren noch für Programmierer in weiten Zügen überblickbar

---

### **Schicht 1 – Physical Layer – Bitübertragungsschicht**

Verfahren und Techniken um elektrische Signale (als 0 und 1 interpretiert) zu übertragen  
z.B. über Strom-Leitungen, Funk, Lichtwellen-Leiter  
dazu gehört Signal-Codierung

zugehörige Geräte: Leitungen, Stecker, Abschlusswiderstände, Repeater, Hubs, Ethernet, Token-Ring,  
zugehörige Protokolle: V.24, RS 423

### **Schicht 2 – Data Link Layer – Sicherungsschicht**

(auch: Abschnittssicherungsebene, Verbindungssicherungsschicht, Verbindungsebene, Prozedurebene, Datensicherungsschicht, ...)

Datenflusskontrolle

regelt zuverlässige und Datensichere Übertragung (Signal-Trennung, Signalstärke, Modulation, ...) → elementare Fehler-Erkennungs-Mechanismen

zugehörige Geräte: Bridge, Switch  
zugehörige Protokolle: ARP, STP, IEEE 802.1 (WLAN-Protokolle)

### **Schicht 3 – Network Layer – Vermittlungsschicht**

(auch: Paketebene, Netzwerkschicht)

sorgt für Leitungs-Verbindung (Adresse zu Adresse) oder Weiterleitung von Daten(Paketen)  
Routing, es werden passende Verbindungen gesucht, Daten-Pakete werden auf geeignete Leitungen vermittelt, Daten(Pakete werden nach Netzwerk-Zugehörigkeit sortiert)  
Datenfluss-Kontrolle

zugehörige Geräte: Router, Layer-3-Switch (BRouter)  
zugehörige Protokolle: IP, X.25, IPsec

### **Schicht 4 – Transport Layer – Transport-Schicht**

auch: Ende-zu-Ende-Kontrolle, Transport-Kontrolle

Zuordnung von bestimmten Datenpaketen zu den zugehörigen Anwendungs-Programmen (über die Ports)

logische Ende-zu-Ende-Verbindung

Multiplex-Verfahren, Fehlersicherungs- und Fehlerbehebungs-Verfahren

Paritäts-Prüfung, CRC-Fehler-Prüfung

zugehörige Geräte:  
zugehörige Protokolle: TCP, UDP, SPX, NetBEUI

---

### **Schicht 5 – Session Layer – Sitzungs-Schicht**

(auch: Kommunikationssteuerungsschicht)

Dienste für organisierten Datenaustausch; Synchronisation des Datenaustausch

Wiederaufsetzungspunkte

Prozess-zu-Prozess-Verbindung; Verbindung zwischen den Endgeräten; Quittierung, Handshake, ...

zugehörige Geräte:

zugehörige Protokolle: NetBIOS

ab hier viele bekannte Internet-Protokolle (HTTP, FTP, SMTP, NNTP)

### **Schicht 6 – Presentation Layer – Darstellungs-Schicht**

(auch: Datendarstellungsschicht, Datenbereitstellungsebene)

Übertragung der Daten in eine für die Kommunikation geeignete Form (z.B. muss ein Baum in eine Sequenz umgewandelt werden)

Umsetzung der System-abhängigen Daten in Kommunikations-Daten und umgekehrt

Übersetzung von Daten-Formaten ineinander; Übertragung der Daten in Standard-Formate

zugehörige Anwendungen: Betriebssystem-Schicht

zugehörige Protokolle: ASN.1

### **Schicht 7 – Application Layer – Anwendungs-Schicht**

Funktion / Prozeduren für die Anwendungen und das Netz-Management der Anwender-Programme, Daten-Ein- und -Ausgabe

zugehörige Anwendungen: Browser, eMail-Client, FTP-Programm, Konsole

zugehörige Protokolle:

### **nach neueren Modellen:**

Layer-8: finanzielle Schicht

Layer-9: politische Schicht

Wenn's notwendig ist: Eselsbrücken zum Lernen der Schichtenfolge

**An dem Sonntag trug Verena nen String in blau.**

**An Darmausgängen sichtete Travis verschiedne sichelförmige Birnen.**

**Alle durstigen Slowenen trinken viel schäumiges Bier.**

**Alle deutschen Schüler trinken verschiedene Sorten Bier.**

**A D S T V S B**

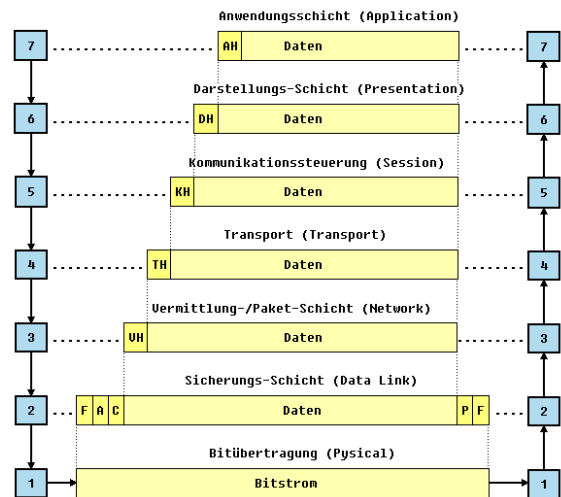
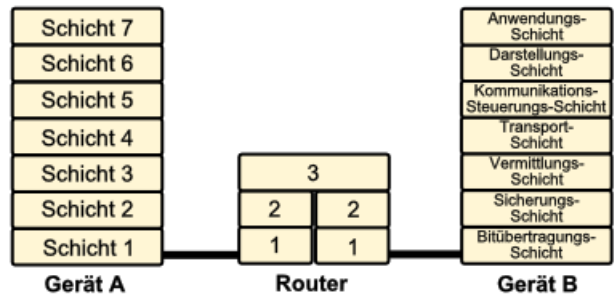


Bei Sender startet die Kommunikation auf einer bestimmten Ebene (Schicht) – meist ist es die Anwendungs-Schicht (Schicht 7). Der Nutzer hat z.B. eine Anfrage an den Browser gestellt (Internetseiten-Aufruf im www).

Die Anfragen werden nun Schicht für Schicht runtertransferiert und auf Schicht 1 (Bitübertragungsschicht) weitergeleitet (physikalisch übertragen). Im Empfänger gehen die Anforderungen den umgekehrten Weg durch die Schichten.

In der Anwendungs-Schicht angekommen, wird eine Antwort generiert und die Kommunikation geht zurück den umgekehrten Weg.

Zwischenstationen – wie im Beispiel ein Router – nutzen nur wenige übereinanderliegende Schichten aus. Sie dienen nur der Weiterleitung und ev. Umsetzung der Signale auf anderen Leitungen / Kommunikationswegen usw. usf.



Q: <http://www.netzmafia.de/skripten/netze/netz0.html#0.1>  
(Prof. Jürgen Plate)

---

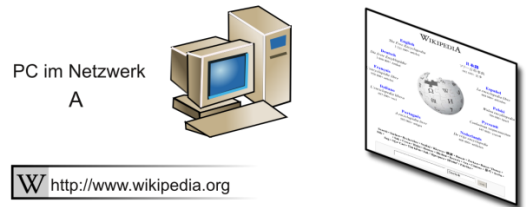
### **Entitäten des OSI-Modells**

- **SAP**      Service Access Point  
Dienst-Zugangspunkt
- **SDU**      Service Data Unit  
Nutzdaten eines Dienstes
- **ICI**      Interface Control Information  
Service-Informationen für einen Dienst
- **PDU**      Protocol Data Unit  
Nutzdaten eines Protokolls
- **PCI**      Protocol Control Information  
Steuer-Informationen für einen Dienst

# OSI-7-Layer-Model (Open Systems Interconnection Reference Model)

Begriffe: Englisch - Deutsch

7 Application Layer	- Anwendungsschicht
6 Presentation Layer	- Darstellungsschicht
5 Session Layer	- Sitzungs- bzw. Kommunikationsschicht
4 Transport Layer	- Transportschicht
3 Network Layer	- Netzwerk- bzw. Vermittlungsschicht
2 Data Link Layer	- Sicherungsschicht
1 Physical Layer	- Bitübertragungsschicht

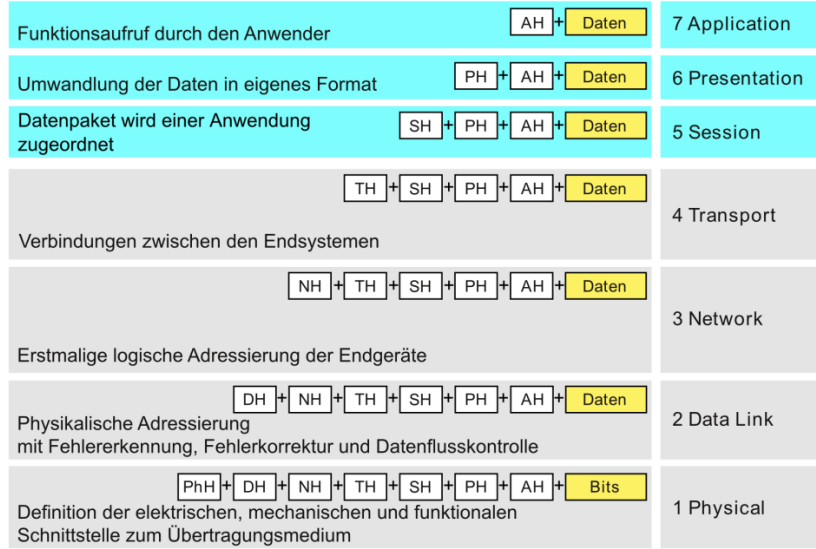


Der Benutzer empfängt lediglich die Antwort des Servers ("wikipedia.org"-Startseite). Im Allgemeinen bekommt er von der Schachtelung seines Seitenaufrufs durch die Ebenen seines PCs (abwärts) und vom Parsen der Antwort des Servers zurück durch die Ebenen seines PCs (aufwärts) nichts mit!

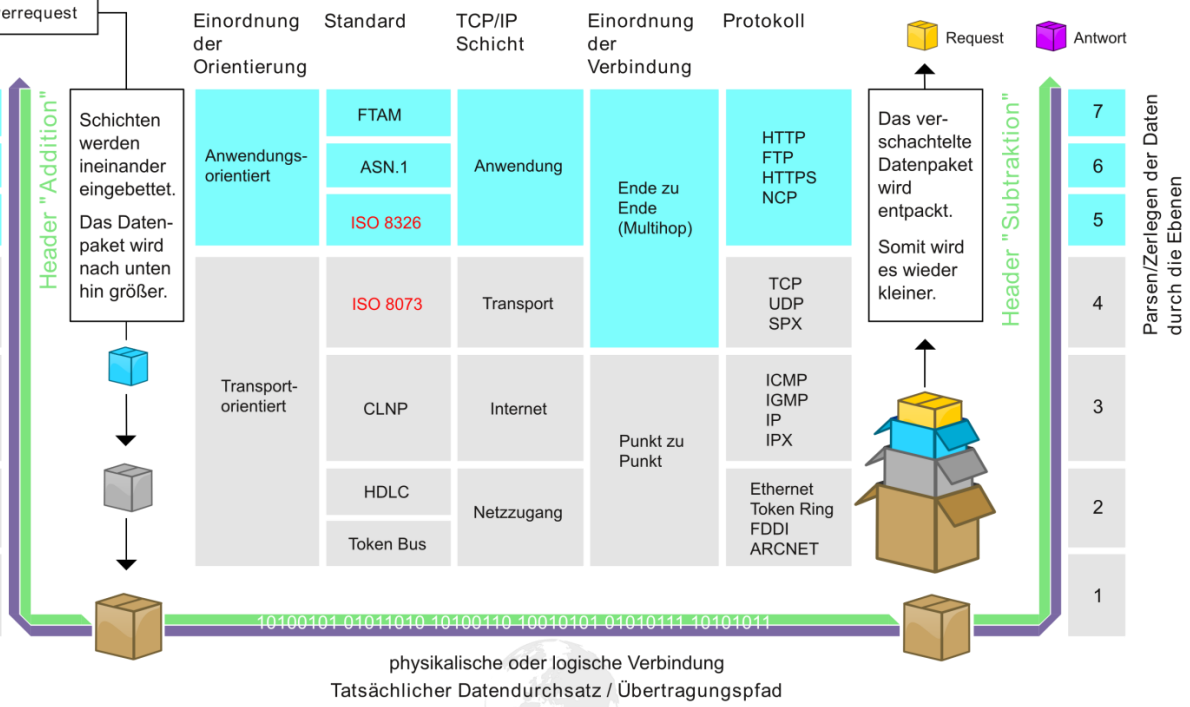
Server im Netzwerk B

Server schickt die entsprechenden Daten über die selbe Methode zurück. (s.u.)

Zusammenbau des Pakets:  
(Package Assembling/Formatting)



Zusammensetzung der Abkürzungen oben:  
Anfangsbuchstabe der Schicht und "H" für Header.  
z.B. Application Header = AH



10100101 01011010 10100110 10010101 01010111 10101011

physikalische oder logische Verbindung  
Tatsächlicher Datendurchsatz / Übertragungspfad

Autor: gob (www.godofbytes.de)  
Bildversion 3.0 (14. Okt. 2006)  
SVG-Neuaufbau + Korrektur: der\_Fahrer  
Dateiversion 3.1 (21. Juni 2010)

Q: de.wikipedia.org (gob)

## **Vereinfachtes Modell für Paketierung im Internet – Waren-Ver- und Entpackung**

Eine Festplatte (moderne SSD) wird in Taiwan hergestellt. Für den Transport in die Verpackungs-Abteilung werden die SSD in ein antistatisches Tütchen gesteckt (Paket 1). In der Verpackungs-Abteilung werden nun diese Pakete (Tütchen mit der SSD) in einen kleinen Karton gesteckt (Paket 2). Dazu kommen noch einige Zusatz-Informationen (Installations-Anleitung, Garantie-Urkunde, Warn.Hinweise, ...). Von diesem Paket 2 werden nun z.B. 100 Stück in ein Paket 3 für die Groß-Händler gesteckt. Einige dieser Kartons werden auf eine Palette positioniert und schön mit Folie umwickelt (Paket 3). Durch die Folie hindurch kann man den Lieferschein erkennen mit der Zieladresse der Palette – es geht nach Deutschland. Die Palette wird mit weiteren Paletten und einer Zoll-Deklaration in einen Schiffs-Container (Paket 4) geschoben. Dieser Container geht mit einem Schiff (in Begleitung vieler anderer Container und anderer Stückgüter) als Paket 5 über die Meere auf den Weg nach Deutschland.

Im Hamburger Hafen wird nun zuerst der Container geöffnet, die Zoll-Unterlagen geprüft und entfernt. Die einzelnen Paletten aus dem Container gehen nun zu den Adressen auf den folierten Lieferscheinen auf den Weg. Beim Groß-Händler werden die Paletten von ihrer Umwicklung befreit und die 100-Stück-Pakete entnommen. Die normalen (Endkunden-)Händler bekommen nun die 100er Pakete und entnehmen die kleinen Einzelpackungen (mit immer einer SSD) für die Verkaufsregale. Der Kund kauft eine Packung und holt zuhause stolz seine erworbene SSD aus der Packung, entfernt die Antistatik-Tüte und hält sie endlich direkt in seiner Hand.

---

## 2.4.x.y. DoD-Modell

vom US Department of Defense (DoD) entwickelte ursprüngliche Schicht-Struktur für den Vorläufer des Internets – das ARPANET

vierschichtig

Schicht 7	Anwendungs-Schicht	Schicht 4	Prozess
Schicht 6	Darstellungs-Schicht		
Schicht 5	Kommunikations-Steuerungs-Schicht		
Schicht 4	Transport-Schicht		
Schicht 3	Vermittlungs-Schicht	Schicht 3	Host-to-Host
Schicht 2	Sicherungs-Schicht	Schicht 2	Internet
Schicht 1	Bitübertragungs-Schicht	Schicht 1	Netzwerk-Zugriff

DoD-Modell (grün)  
im Vergleich zum ISO-OSI-Modell

- **Process**                    Anwendungen / Nutzer-Programme
- **Host-to-Host**            Ablaufsteuerung der Kommunikations-Prozesse
- **Internet**                    Vermittlung und Kommunikation zwischen verschiedenen Hosts (Rechnern)
- **Network Access**        Zugriff auf Übertragungsmedien

## 2.4.x.y. TCP/IP-Referenz-Modell

historisch gewachsen

Vorteile:

Praxis-orientiert

OpenSource

keine Lizenz-Gebühren

praktisch überall genutzt

Nachteile:

ungünstige / ungleichmäßige (z.T. auch keine exakte) Trennung der Schichten



analoge Ebenen im Vergleich: ISO-OSI-, DoD- (grün) und TCP/IP-Modell (blau)

Ports für TCP und UDP

sind Zuordnungen von Protokollen zu bestimmten Anwendungs-Programmen

Port	Service-Name	Beschreibung
<b>UDP</b>		
53	DNS	
<b>TCP</b>		
21	FTP	
23	Telnet	
80	HHTTP	

---

## IP-Schicht

IP ... Internet-Protokoll

binäre Adresse eines Netzwerk-Teilnehmers

Adresse im Internet – also Adresse innerhalb des / eines Netzes

echte und unverwechselbare Adresse einer Netzwerk-Station ist die MAC-Adresse (Media-Access-Control)

wird von den Herstellern fest eingebaut, technisch aber veränderbar, individuelle Adresse eines Netzwerk-Gerätes

wird bei der Datenübertragung immer mit übertragen (von Sender und Empfänger)

deshalb auch als physikalische Adresse bezeichnet

praktisch eine 48 bit-Zahl, die hexadezimal notiert wird

aus der aktuellen IP-Adresse eines Nutzers kann man ermitteln:

In welcher Region (bis hin zu Städten) befindet sich der User?

Welcher Internet-Provider betreut den User?

der Provider führt – vorrangig für Abrechnungszwecke – Protokolle, aus denen zu entnehmen ist:

Welche Seiten hat der Nutzer, wie lange aufgerufen?

Was hat er heruntergeladen?

Wonach hat der Nutzer gesucht?

Welche Seite wurde als nächstes aufgerufen?

## Exkurs: MAC-Adresse

48 bit-Hardware-Adresse eines Netz-Gerätes (6 Byte)

24 bit (3 Byte) Organizationally Unique Identifier

24 bit (3 Byte) Network Interface Controller Specific

gehört zu Layer 2 (ISO-OSI: )

kanonische Schreibweisen:

xx-xx-xx-xx-xx-xx oder xx:xx:xx:xx:xx:xx

aber auch andere Schreibungen – z.B. auf Aufklebern von Geräten zu finden:

xxxxxxxxxxxx

xxxx.xxxx.xxxx

Hersteller	MAC-Adresse(n)
Intel	00:07:E9:xx:xx:xx
Asus	00:15:F2:xx:xx:xx
Compaq	00:50:8B:xx:xx:xx
Cisco	00:60-2F:xx:xx:xx

Daten-Q: de.wikipedia.org

Abfrage in Windows:

Win 95 – ME: winipcfg

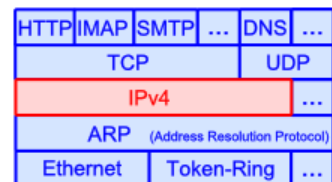
ab Win 2000: ipconfig /all

ev. auch: getmac /v

Android: "Einstellungen" "Telefoninfo" "Hardware" "Informationen"

iOS: "Einstellungen" ""Allgemein" "Info" "Wi-Fi-Adresse"

Linux: ip link



Paket-Vermittlung

Verbindungs-los

realisiert / organisiert Wege-Wahl

Schutz vor Überlastung einzelner Medien

Paket-Verfolgung

→ <https://traceroute-online.com>



## IP-Adresse

Identifikation eines Knoten

Festlegung von Ziel und für die Rück-Antwort auch für die Quelle  
quasi die Post-Anschrift für die Datenpakete

Netzwerk-Maske

trennt Rechner-Netze untereinander

Internet-Adressen verschleiern MAC-Adressen

machen es möglich, dass Geräte unabhängig von ihrer aktuellen Position im Netz erreichbar sind

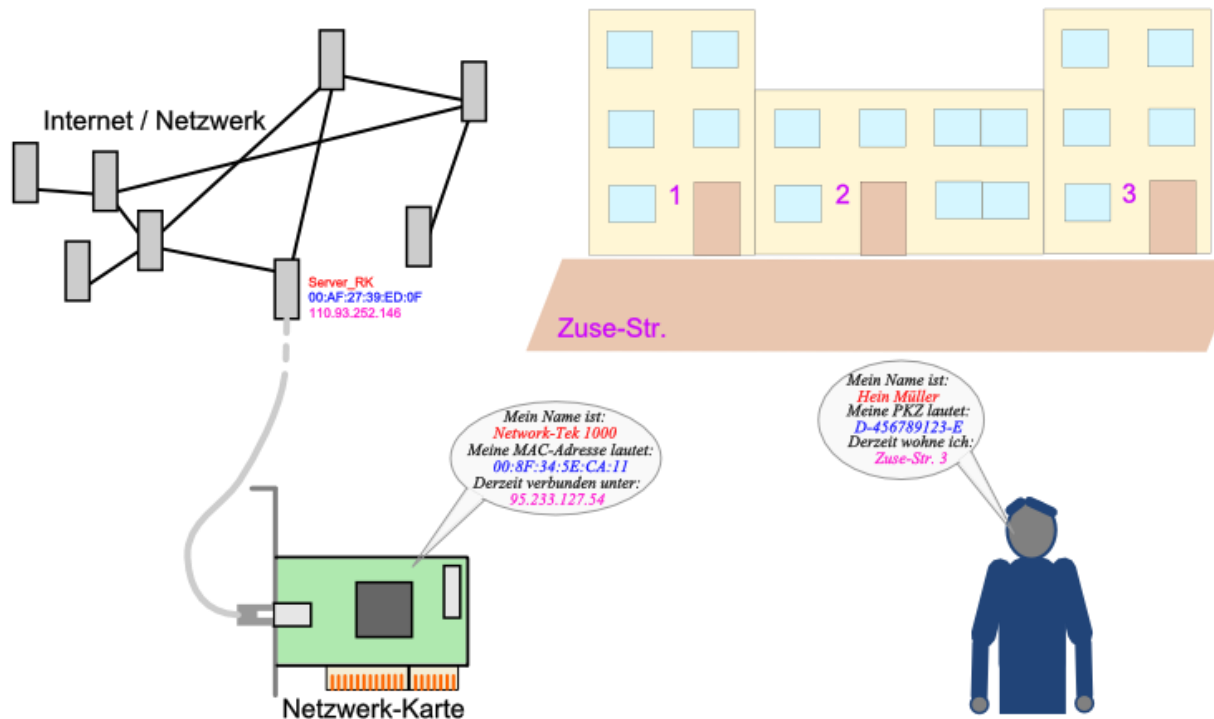
## IP-Adresse Version 4 (IPv4)

seit 1981 existierende Adressen-Angabe für die derzeitige Einbindung ins Netz  
besteht aus vier dezimal geschriebene 8-bit-Zahlen, die mittels Punkt voneinander getrennt  
notiert werden, z.B. 177.39.238.184

quasi eine 32-bit-Adresse

theoretisch stehen damit  $FF.FF.FF.FF_H$  also  $4'294'967'295_D$  Anschlüsse / Adressen zur Verfügung

trotz Aktualisierung und weitestgehender Umstellung auf IPv6 (→ ) immer noch das Standard-Adress-Protokoll im Internet (das geht immer)



---

die Ausnutzung des gesamten Adress-Bereiches wirft aber einige Probleme auf

- Wer vergibt die Adressen?
- Was passiert, wenn ein PC ausgetauscht wird? Bekommt er eine neue Adresse? Ist der Nutzer dann noch erreichbar?
- Wie kann man die Geräte eines kleineren Subnetzes informieren / verwalten / ...?
- Wie lassen sich Computer sicher schützen, wenn sie direkt aus dem Internet erreichbar sind?
- Wie kann ein (Internet-)Router erkennen, was im eigenen Netz verbleiben soll und was ins Internet gehen soll?

Die Netzwerk-Administratoren wollen die – ihnen zugeordnet – Rechner eigenständig verwalten.

Firmen brauchen praktisch nur eine oder wenige Internet-Adressen, aber relativ viele Adressen für die Arbeitsrechner.

Insgesamt würde die Zahl der Adressen nicht ausreichen.

Durchsuchen des gesamten Adress-Bereiches würde ewig dauern.

Braucht man nur 1 ms pro Adresse (real sind 100 – 1000 ms eher realistisch), dann würde ein vollständiger Scan  $4'294'967'295 \text{ ms} = 4'294'967,3 \text{ s} = 71582,8 \text{ min} = 1193,0 \text{ h} = 49,7 \text{ d}$  dauern. Unter Real-Bedingungen also Jahre. Da hat sich die Adress-Nutzung längst schon wieder geändert.

Lösung sind einzelne Blöcke im IP-Adressbereich (Adress-Gruppen).

ein Teil der Adressen wird zentral verwaltet

damit ist z.B. Zuordnung zu Ländern (z.B.: → \*.de od. \*.ch) oder bestimmten Strukturen usw. möglich

z.B. Abgrenzung der Rechner des Militärs (→ \*.mil), Regierung der USA (→ \*.gov), Lehrinrichtungen (→ \*.edu) und der Wirtschaft (→ \*.com)

Achtung! Die Zuordnung zu den Text-Adressen soll hier nur inhaltlich verstanden werden.

Durch Fehleinschätzung des Bedarfs an Adressen wurden Einrichtungen viel zu große Adressbereiche zugeordnet. Viele Adressen können gar nicht genutzt werden, weil sie für Einrichtungen reserviert sind, die diese gar nicht realisieren können (- auch zukünftig nicht).

Für andere Bereiche sind zu kleine Adressbereiche vorgesehen worden.

Vergabe früher ausschließlich durch IANA (Internet Assigned Numbers Authority)

IBM z.B. 9.0.0.0/8

debis AG (Daimler-Benz-Tochterunternehmen) 53.0.0.0/8 einzige deutsche Firma mit einem Klasse-A-Netz

heute von regionalen Organisation, die sich weitgehend an Kontinenten orientieren

häufige IPs für Europa: 80.x.x.x, 192.x.x.x, 193.x.x.x, 194.x.x.x

Ein anderer Teil der Adressen klar von der zentralen Vergabe ausgeschlossen. Sie sind für private / Firmen- / Instituts-Zwecke reserviert. Adress-Aufrufe werden innerhalb dieser Netze nicht nach außen weitergeleitet.

		mögliche Hosts	geeignet für ...			
Class A, privates Internet	10.x.x.x	16'777'214	Staaten			
Class B, privates Internet	172.16.x.x → 172.31.x.x	1'048'574	Universitäten, Groß-Konzerne			
Class C, privates Internet	192.168.x.x	254	Private Netze Büros, Kleinfirmen			

Rechner untereinander über Switche verbunden Die verteilen Daten über die MAC-Adressen (ISO-OSI Schicht 2). Switche kennen keine IP-Adressen. Nutzung des Internet-Protokolls zwischen den Hots nur der Einfachheit halber. Es könnte auch ein anderes Protokoll genutzt werden (z.B. IPX/SPX (Netware-Netze)).

Innerhalb des privaten Netzes werden Adressen entweder **statisch** vergeben (Adminsitrator muss dann den Überblick behalten oder **dynamisch** verteilt. Ein Gerät – meist der Router ist dann ein sogenannter **DHCP**-Server (Dynamic Host Configuration Protocol), der beim Rechner-Start auf Anfrage durch den Rechner (fungiert als DHCP-Client) diesem eine IP-Adresse aus einem definierten Bereich vergibt. Häufig sind die Router die DHCP-Server in den privaten Netzen. In Firmen-Netzen mit anderen Servern haben meist diese auch die DHCP-Funktion.

DHCP liefert die IP-Adresse, eine Netzmaske, den Gateway (ins Internet) und den DNS-Server

Sollen diese Netze auch ins Internet zugreifen können, wird ein Router (ISO-OSI Schicht 3) gebraucht, der (mindestens) eine echte – zentral vergebene – Internet-Adresse haben muss und (mindestens) eine Adresse aus dem privaten Netzwerk

Die äußeren Adressen werden entweder direkt gekauft oder über einen Internet-Povider (ISP, Internet-Service-Provider) zeitweise gemietet. Typisch für die häusliche bzw. Kleinfirmen-Internet-Anbindung.

Man erhält typischerweise vom ISP eine Adresse für 24 h.

Abfragbar z.B.

in Windows: über Konsole (cmd) und dort ipconfig /all

im Browser: <http://www.meineip.de>

### **böse Frage zwischendurch:**

***Könnte ich meinem Rechner für mein privates, aber über einen Router ins Internet angeschlossene Netz, die Adresse 10.16.192.24 geben, ohne mit Problem (IP-Konflikten) rechnen zu müssen?***

Bestimmte Adressen dienen speziellen Zwecken

keine Adressen 0.x.x.x oder 127.x.x.x

127.0.0.1 ist immer der eigene Netzwerk-Anschluss selbst (Loop)  
damit immer Test der Netzwerk-Adresse möglich

169.254.0.0/16

ab 224.x.x.x für Multicast reserviert  
 Versand / Information von Gruppen von Empfängern

Klassen-Kennung + Präfix + Suffix

Klasse	Kenn-Bits	Bits im Präfix	max. Anzahl mögl. Netze	Bits im Suffix	max. Anzahl mögl. Hosts pro Netz
A	0	7	128	24	$2^{24} - 2$
B	10	14	16'384	16	$2^{16} - 2$
C	110	21	2'097'152	8	$2^8 - 2$

diverse Adressen noch reserviert für verschiedenste / zukünftige Zwecke

CIDR-Notation (Classless Inter-Domain Routing)  
 statt der Netzmaske wird hinter der IP-Adresse und einem Schrägstrich die Anzahl der 1-Bits geschrieben  
 also bei 255.255.255.0 → /24

Netzwerk-Adresse:	dez	192.168.0.34	
Netz-Maske:	dez	255.255.255.0	
	dez	/24	CIDR
Netzwerk-Adresse:	bin	1100 0000.1010 1000.0000 0000.0010 0010	
Netz-Maske:	bin	1111 1111.1111 1111.1111 1111.0000 0000	
Netz-Teil:	bin	1100 0000.1010 1000.0000 0000.	
Netz-Teil:	dez	192.168.0.	
Host:	bin	.0010 0010	
Host-Teil:	dez	.34	

### Netz-Adresse

Berechnung der Netz-Adresse = IP-Adresse UND Netzmaske  
 Netz-Adresse auch NID (Netz-ID)

Netzwerk-Adresse:	bin	1100 0000.1010 1000.0000 0000.0010 0010
Netz-Maske:	bin	UND 1111 1111.1111 1111.1111 1111.0000 0000
Netz-Adresse:	bin	1100 0000.1010 1000.0000 0000.0000 0000

entspricht:

Netz-Adresse:	dez	192.168.0.0
---------------	-----	-------------

## Host-Adresse

Berechnung Host-Adresse = IP-Adresse UND NICHT Netzmaske  
 Host-Adresse = Host-Nummer

Netzwerk-Adresse:	bin	1100 0000.1010 1000.0000 0000.0010 0010
NICHT Netz-Maske:	bin	UND 0000 0000.0000 0000.0000 0000.1111 1111
Host-Adresse:	bin	0000 0000.0000 0000.0000 0000.0010 0010

entspricht:

Host-Adresse:	dez	<b>192.168.0.34</b>
---------------	-----	---------------------

## Anzahl Hosts

Berechnung der möglichen Host-Stationen =  $2^{(32 - \text{Netzmaske-Bits})} - 2$

Host-Anzahl:	dez	$2^{(32-24)} - 2 = 2^8 - 2 = 256 - 2 = 254$
--------------	-----	---------------------------------------------

bzw. über den direkten Weg:

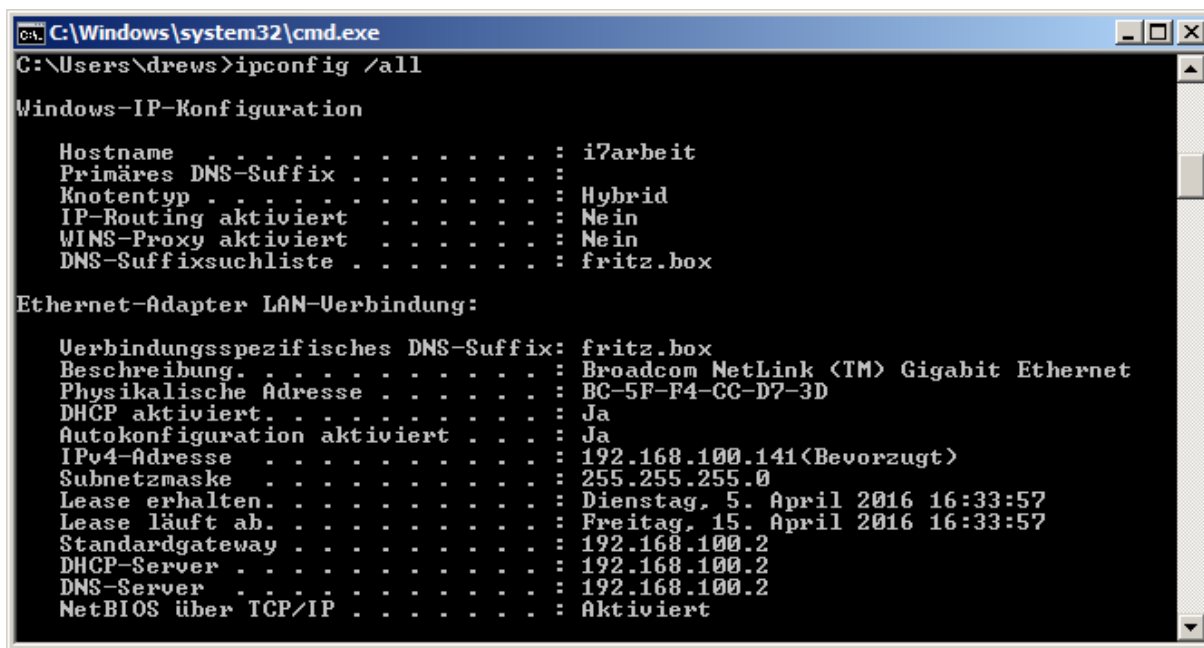
Adressbereich bis:	bin	.1111 1111
Adressenanzahl	dez	<b>256</b>
Host-Anzahl:	dez	$256 - 2 = 254$

Um alle relevanten Daten zum Host und seinem Netz zu ermitteln kann folgendes Schema verwendet werden:

Netzwerk-Adresse:	dez	<b>192.168.0.34</b>	
Netz-Maske:	dez	<b>255.255.255.0</b>	
	dez	<b>/24</b>	CIDR
Netzwerk-Adresse:	bin	1100 0000.1010 1000.0000 0000.0010 0010	
Netz-Maske:	bin	1111 1111.1111 1111.1111 1111.0000 0000	
Netz-Teil:	bin	1100 0000.1010 1000.0000 0000.	
Netz-Teil:	dez	<b>192.168.0.</b>	
Host:	bin	.0010 0010	
Host-Teil:	dez	<b>.34</b>	
<b>Host-Adressen:</b>			
Adressbereich von:	bin	.0000 0000	
Netzname:	bin	1100 0000.1010 1000.0000 0000.0000 0000	
Netzname:	dez	<b>192.168.0.0</b>	
Adressbereich bis:	bin	.1111 1111	
Adressenanzahl	dez	<b>256</b>	
Broadcast:	bin	1100 0000.1010 1000.0000 0000.1111 1111	
Broadcast:	dez	<b>192.168.0.255</b>	
Host-Anzahl:	dez	$256 - 2 = 254$	
1. Host (Nr.):	bin	0000 0001	
1. Host (Nr.):	dez	<b>1</b>	immer so
1. Hostadresse:	bin	1100 0000.1010 1000.0000 0000.0000 0001	
1. Hostadresse:	dez	<b>192.168.0.1</b>	
n. Host (Nr.):	bin	1111 1110	
n. Host (Nr.):	dez	<b>254</b>	
n. Hostadresse:	bin	1100 0000.1010 1000.0000 0000.1111 1110	
n. Hostadresse:	dez	<b>192.168.0.254</b>	

## IP-Rechner in Netz:

<http://www.trinler.net/de/service/tools/ipcalc.html>



```
C:\Windows\system32\cmd.exe
C:\Users\drews>ipconfig /all

Windows-IP-Konfiguration

    Hostname . . . . . : i7arbeit
    Primäres DNS-Suffix . . . . . :
    Knotentyp . . . . . : Hybrid
    IP-Routing aktiviert . . . . . : Nein
    WINS-Proxy aktiviert . . . . . : Nein
    DNS-Suffixsuchliste . . . . . : fritz.box

Ethernet-Adapter LAN-Verbindung:

    Verbindungsspezifisches DNS-Suffix: fritz.box
    Beschreibung. . . . . : Broadcom NetLink (TM) Gigabit Ethernet
    Physikalische Adresse . . . . . : BC-5F-F4-CC-D7-3D
    DHCP aktiviert. . . . . : Ja
    Autokonfiguration aktiviert . . . : Ja
    IPv4-Adresse . . . . . : 192.168.100.141(Bevorzugt)
    Subnetzmaske . . . . . : 255.255.255.0
    Lease erhalten. . . . . : Dienstag, 5. April 2016 16:33:57
    Lease läuft ab. . . . . : Freitag, 15. April 2016 16:33:57
    Standardgateway . . . . . : 192.168.100.2
    DHCP-Server . . . . . : 192.168.100.2
    DNS-Server . . . . . : 192.168.100.2
    NetBIOS über TCP/IP . . . . . : Aktiviert
```

ursprüngliche Adressierung / classful routing

**Netzwerk-Klassen (IPv4)**

Netzwerk-Klasse	Netz-Anteil	Host-Anteil	IP-Bereich / -Adresse	Netzwerk-Maske	Anzahl Adresse	Anzahl Subnetze	
<b>A</b>	8 bit	24 bit	10.0.0.0 → 10.255.255.255	01111111 00000000 00000000 00000000 255.0.0.0 /8	16'777'216	128	weltweite / überregionale Netze
<b>B</b>	16 bit	16 bit	172.16.0.0 → 172.31.255.255 169.254.0.0 → 169.254.255.255	10111111 11111111 00000000 00000000 255.255.0.0 /16	65'536	16'384	Standard Firmen-Netze
<b>C</b>	24 bit	8 bit	192.168.0.0 → 192.168.255.255	11011111 11111111 11111111 00000000 255.255.255.0 /24	256	2'097'152	Standard Heim- und Firmen-Netze (klein)
<b>D</b>	4 bit	28 bit		11100000 00000000 00000000 00000000 224.0.0.0 /4			für Multicast-Gruppen
<b>E</b>	4 bit	28 bit		11110000 00000000 00000000 00000000 240.0.0.0 /4			für zukünftige Zwecke reserviert

abgelöst 1993 durch Klassen-lose IP-Adressierung (CIDR)  
praktisch breite Zurverfügungstellung fast des gesamten Adressbereiches

## Klassen-lose Netzwerk-Adressen (CIDR)

Netzwerk-Klassen sind praktisch immer überdimensioniert  
 dazu kam historisch bedingte und wenig durchdachte Zuordnung der Netzwerk-Klassen  
 Universität von Berkeley hat ein Klasse A-Netz aber nicht mal annähernd 16 Mill. Rechner

CIDR wurde 1993 eingeführt  
 CIDR .. Classless Interdomain Routing  
 lässt feine Aufteilung der Netze zu

Einführung der variablen Netz-Maske (Subnetz-Maske)  
 Schachtelung in Subnetze möglich, die für sich optimale (minimale) Größe haben

Netzwerk-Adresse:	dez	<b>192.168.190.156</b>	
Netz-Maske:	dez	<b>255.255.224.0</b>	
	dez	<b>/19</b>	CIDR
Netzwerk-Adresse:	bin	1100 0000.1010 1000.1011 1110.1001 1100	
Netz-Maske:	bin	1111 1111.1111 1111.1110 0000.0000 0000	
Netz-Teil:	bin	1100 0000.1010 1000.1010 0000	
Netz-Teil:	dez	<b>192.168.160</b>	
Host:	bin	0001 1110.1001 1100	
Host-Teil:	dez	<b>30.156</b>	
<b>Host-Adressen:</b>			
Adressbereich von:	bin	0 0000.0000 0000	
Netzname:	bin	1100 0000.1010 1000.1010 0000.0000 0000	
Netzname:	dez	<b>192.168.0.0</b>	
Adressbereich bis:	bin	1 1111.1111 1111	
Adressenanzahl	dez	<b>8192</b>	
Broadcast:	bin	1100 0000.1010 1000.1011 1111.1111 1111	
Broadcast:	dez	<b>192.168.191.255</b>	
Host-Anzahl:	dez	<b>8192 - 2 = 8190</b>	
1. Host (Nr.):	bin	1010 0000 0000 0001	
1. Host (Nr.):	dez	<b>1</b>	immer so
1. Hostadresse:	bin	1100 0000.1010 1000.0000 0000.0000 0001	
1. Hostadresse:	dez	<b>192.168.160.1</b>	
n. Host (Nr.):	bin	1011 1111.1111 1110	
n. Host (Nr.):	dez	<b>8191</b>	
n. Hostadresse:	bin	1100 0000.1010 1000.1011 1111.1111 1110	
n. Hostadresse:	dez	<b>192.168.191.254</b>	



Netzwerk-Adresse:	dez	<b>192.168.222.24</b>	
Netz-Maske:	dez	<b>255.255.255.240</b>	
	dez	<b>/28</b>	CIDR
Netzwerk-Adresse:	bin	1100 0000.1010 1000.1101 1110.0001 1000	
Netz-Maske:	bin	1111 1111.1111 1111.1111 1111.1111 0000	
Netz-Teil:	bin	1100 0000.1010 1000.1101 1110.0001 0000	
Netz-Teil:	dez	<b>192.168.222.16</b>	
Host:	bin	. 0000 1000	
Host-Teil:	dez	<b>.8</b>	
<b>Host-Adressen:</b>			
Adressbereich von:	bin	.0000 0000	
Netzname:	bin	1100 0000.1010 1000.1101 1110.0001 0000	
Netzname:	dez	<b>192.168.222.16</b>	
Adressbereich bis:	bin	.0000 1111	
Adressenanzahl	dez	<b>16</b>	
Broadcast:	bin	1100 0000.1010 1000.1101 1110.0001 1111	
Broadcast:	dez	<b>192.168.222.31</b>	
Host-Anzahl:	dez	<b>16 - 2 = 14</b>	
1. Host (Nr.):	bin	.0000 0001	
1. Host (Nr.):	dez	<b>1</b>	immer so
1. Hostadresse:	bin	1100 0000.1010 1000.1101 1110.0001 0001	
1. Hostadresse:	dez	<b>192.168.222.17</b>	
n. Host (Nr.):	bin	.0000 1110	
n. Host (Nr.):	dez	<b>14</b>	
n. Hostadresse:	bin	1100 0000.1010 1000.1101 1110.0001 1110	
n. Hostadresse:	dez	<b>192.168.222.30</b>	

### Berechnungen der Netzwerk-Adressen-Teile:

(direkte) **Broadcast-Adresse** = Netzwerk-Adresse ODER NICHT Netz-Maske

**Netz-Adresse** = Netzwerk-Adresse UND Netz-Maske

**Host-Adresse** = Netzwerk-Adresse UND NICHT Netz-Maske

jeder Netzwerk-Architekt / -Administrator kann / sollte jetzt eine für seine Verhältnisse / Firma / Einrichtung passende Netzmaske auswählen, um die Adressen optimal auszunutzen  
immer ein paar Adressen Reserve einplanen

aus meiner Sicht als Adminstrator: wenn es die Klassen-lose Adressierung sein soll, dann ungefähr die Hälfte bis die gleiche Anzahl aktuell vorhandener Hosts hinzufügen  
nichts ist grausiger, als alles von vorne neu zu adressieren

traditionell bleiben die meisten Admins bei den Netzwerk-Klassen

Nummern lassen sich schnell merken, können immer gleich in verschiedenen Netzen angewendet werden, großer Erfahrungs-Schatz → geringere Fehler-Quote  
ist auch mein Arbeiten

üblicherweise sind die kleinen Admins nur Klasse C-Netz-Betreuer

echte Admins haben größere Netze, da wird erfahrungsgemäßig sowieso anders gearbeitet

## Berechnungen der optimalen Netz- / Subnetz-Maske:

gegeben: zugewiesenes Netz 194.123.0.0/23 (von ICANN)

Einrichtung	benötigte Hosts	Teilnetz	
Haus 1	123	A	
Haus 2	12		
Haus 3	34	B	
Haus 4	77	C	

### Algorithmus zum Berechnen:

1. Sortieren der Teilnetze nach Hostanzahl → $HA_x$	$HA_A = 135; HA_B = 34; HA_C = 77$  $HA_1 = 135$ $HA_2 = 77$ $HA_3 = 34$
2. Hostanzahl um 2 erhöhen (für Netzwerk + Broadcast) → $HN_x = HA_x + 2$	$HN_1 = 137$ $HN_2 = 79$ $HN_3 = 36$
3. nächsthöhere Größe $2^n$ suchen → $HAB_x$ $n_x = 1 + \text{ABRUNDEN}(\log HN_x / \log 2)$ $HAB_x = 2^n$	$n_1 = 8 \rightarrow HAB_1 = 256$ $n_2 = 7 \rightarrow HAB_2 = 128$ $n_3 = 6 \rightarrow HAB_3 = 64$
4. max. Hostanzahl $HM_x$ berechnen (abziehen von Netzwerk und Broadcast) $HM_x = HAB_x - 2$	$HM_1 = 254$ $HM_2 = 126$ $HM_3 = 62$
5. CIDR für Subnetz x berechnen → $CDIR_x$ $CDIR_x = 32 - n_x$	$CDIR_1 = 24$ $CDIR_2 = 25$ $CDIR_3 = 26$
6. Subnetz 1: Netz-Adresse → $NID_1$ erste freie Adresse	$NID_1 = 194.123.0.0/24$
7. erste IP im 1. Subnetz → $IP_{11} = NID_1 + 1$	$IP_{11} = 194.123.0.1$
8. letzte IP im 1. Subnetz → $IP_{m1}$ $IP_{m1} = NID_1 + HN_{x1}$	$IP_{m1} = 194.123.0.254$
9. Broadcast-IP im 1. Subnetz → $BC_1$ $BC_1 = IP_{m1} + 1$	$BC_1 = 194.123.0.255$
10. Subnetz 2: Netz-Adresse → $NID_2$ $NID_{x+1} = BC_x + 1$	$NID_2 = 194.123.1.0/25$
11. äquivalent zu 7: $IP_x = NID_x + 1$	$IP_{21} = 194.123.1.1$
12. letzte IP im 1. Subnetz → $IP_{mx}$ $IP_{mx} = NID_x + HN_{x1}$	$IP_{21} = 194.123.1.126$
13. Broadcast-IP im 1. Subnetz → $BC_1$ $BC_x = IP_{mx} + 1$	$BC_2 = 194.123.1.127$
14. Subnetz 2: Netz-Adresse → $NID_{x+1}$ $NID_{x+1} = BC_x + 1$	$NID_3 = 194.123.1.128$
15. ... bis alle Subnetze (x) abgearbeitet sind	...

### Netze der Klassen-losen Adressierung

Not.	Netz-Maske		Adressen, gesamt	nutzbare Adressen	Bemerkungen
	dezimal Not.	binäre Notierung			
/0	0.0.0.0	0000 0000.0000 0000.0000 0000.0000 0000	4'294'967'296	intern Einteilung in Subnetze	
/1	128.0.0.0	1000 0000.0000 0000.0000 0000.0000 0000	2'147'483'648		
/2	192.0.0.0	1100 0000.0000 0000.0000 0000.0000 0000	1'073'741'824		
/3	224.0.0.0	1110 0000.0000 0000.0000 0000.0000 0000	536'870'912		
/4	240.0.0.0	1111 0000.0000 0000.0000 0000.0000 0000	268'435'456		
/5	248.0.0.0	1111 1000.0000 0000.0000 0000.0000 0000	134'217'728		
/6	252.0.0.0	1111 1100.0000 0000.0000 0000.0000 0000	67'108'864		
/7	254.0.0.0	1111 1110.0000 0000.0000 0000.0000 0000	33'554'432		
<b>/8</b>	255.0.0.0	1111 1111.0000 0000.0000 0000.0000 0000	16'777'216	16'777'214	ehem. Class A
/9	255.128.0.0	1111 1111.1000 0000.0000 0000.0000 0000	8'388'608	8'388'606	
/10	255.192.0.0	1111 1111.1100 0000.0000 0000.0000 0000	4'194'304	4'194'302	
/11	255.224.0.0	1111 1111.1110 0000.0000 0000.0000 0000	2'097'152	2'097'150	
/12	255.240.0.0	1111 1111.1111 0000.0000 0000.0000 0000	1'048'576	1'048'574	
/13	255.248.0.0	1111 1111.1111 1000.0000 0000.0000 0000	524'288	524'286	
/14	255.252.0.0	1111 1111.1111 1100.0000 0000.0000 0000	262'144	262'142	
/15	255.254.0.0	1111 1111.1111 1110.0000 0000.0000 0000	131'072	131'070	
<b>/16</b>	255.255.0.0	1111 1111.1111 1111.0000 0000.0000 0000	65'536	65'534	ehem. Class B
/17	255.255.128.0	1111 1111.1111 1111.1000 0000.0000 0000	32'768	32'766	
/18	255.255.192.0	1111 1111.1111 1111.1100 0000.0000 0000	16'384	16'382	
/19	255.255.224.0	1111 1111.1111 1111.1110 0000.0000 0000	8'192	8'190	
/20	255.255.240.0	1111 1111.1111 1111.1111 0000.0000 0000	4'096	4'094	
/21	255.255.248.0	1111 1111.1111 1111.1111 1000.0000 0000	2'048	2'046	
/22	255.255.252.0	1111 1111.1111 1111.1111 1100.0000 0000	1'024	1'022	
/23	255.255.254.0	1111 1111.1111 1111.1111 1110.0000 0000	512	510	
<b>/24</b>	255.255.255.0	1111 1111.1111 1111.1111 1111.0000 0000	256	254	ehem. Class C
/25	255.255.255.128	1111 1111.1111 1111.1111 1111.1000 0000	128	126	
/26	255.255.255.192	1111 1111.1111 1111.1111 1111.1100 0000	64	62	
/27	255.255.255.224	1111 1111.1111 1111.1111 1111.1110 0000	32	30	
/28	255.255.255.240	1111 1111.1111 1111.1111 1111.1111 0000	16	14	
/29	255.255.255.248	1111 1111.1111 1111.1111 1111.1111 1000	8	6	
/30	255.255.255.252	1111 1111.1111 1111.1111 1111.1111 1100	4	2	Verbindungsnetz zwischen 2 Routern
/31	255.255.255.254	1111 1111.1111 1111.1111 1111.1111 1110	2	1	Sonderfall; Point-to-Point-Verbindungen
/32	255.255.255.255	1111 1111.1111 1111.1111 1111.1111 1111	1	1	Sonderfall; einzelner Host

DNS-Service (DNS = Domain Name-System) zum Umgehen der IP-Adressen  
Umsetzung von IP-Adresse (172.217.18.131) in Text-Adresse (www.google.de)

Weg-Verfolgung im Internet  
unter Windows: in der Konsole (cmd) mittels `tracert adresse`  
adresse kann dabei eine IP-Nummer, eine Internet-Adresse oder eine lokale Netzwerk-Adresse bzw. –Ressource sein

```
C:\Windows\system32\cmd.exe
C:\Users\drews>tracert www.google.de

Routenverfolgung zu www.google.de [172.217.18.131] über maximal 30 Abschnitte:

 1  <1 ms    <1 ms    <1 ms    fritz.box [192.168.100.2]
 2   9 ms     9 ms     9 ms     87.186.225.122
 3  10 ms    11 ms    11 ms    87.186.196.62
 4   9 ms    11 ms    11 ms    hh-ea7-i.HH.DE.NET.DTAG.DE [62.154.33.6]
 5  21 ms    21 ms    21 ms    80.150.170.98
 6  14 ms    14 ms    14 ms    209.85.249.124
 7  15 ms    15 ms    15 ms    72.14.233.216
 8  31 ms    27 ms    27 ms    209.85.249.56
 9  29 ms    28 ms    40 ms    209.85.253.181
10  29 ms    29 ms    29 ms    209.85.253.149
11  28 ms    27 ms    27 ms    arn02s05-in-f3.1e100.net [172.217.18.131]

Ablaufverfolgung beendet.

C:\Users\drews>
```

zur Nutzung des heutigen Internets auf der Basis von IPv4 ist Network Address Translation (NAT) notwendig, um die überzähligen Endgeräte ins Netz einzubinden

NAT ist ein Umschreiben der Adresse in den IP-Headern, dabei ist vorrangig die Absender-Adresse gemeint (Adresse im privaten Netz in die öffentliche Adresse im Providernetz)

Probleme tauchen auf beim Zugriff von außen auf Geräte im privaten Netz, z.B. bei Heimsteuerung (smart home), VPN-Verbindungen, Voice over IP (VoIP) oder Multimedia-Übertragungen

Problem entsteht dadurch, dass sich mehrere Endnutzer beim Provider eine IPv4-Adresse teilen (CGNAT, Carrier Grade NAT), damit keine eindeutige Zuordnung der zurückkommenden IP-Pakete möglich; CGNAT beinhaltet eine zweifache Adress-Umschreibung, wofür die meisten Internet-Protokolle (stammen ja auch meist aus den Anfangsstunden des Internets) nicht klar kommen

Konsequenz ist die Sperrung des Zugriffs von außen beim CGNAT

so z.B. für die Heimsteuerung immer ein Cloud-Account notwendig  
daraus ergeben sich Probleme beim Datenschutz (Wo steht der Server? Welche Datenschutz-Regeln gelten?)

---

## **IP-Adresse Version 6 (IPv6)**

(dieser Abschnitt basiert sehr stark auf dem open-hpi-Kurs "IPv6 in modernen Netzwerken" von Prof. W. BOEDDINGHAUS (Juni/Juli 2018); Hasso Plattner Institut  
Kurs als Selbststudien-Version unter: <https://open.hpi.de/>)

### **Gründe für die Ablösung von IPv4 durch IPv6**

- Protokoll technisch veraltet (seit 1983 in Benutzung)
- historisch gemachte ungünstige Vergabe großer Adress-Bereiche
- technisch begründet lassen sich bestimmte Adressen / Adress-Bereiche nicht nutzen
  - 224.0.0.0 für Multicast reserviert
  - Adressen über 240.0.0.0 nicht vorgesehen
- außer für Afrika keine freien Adressen mehr verfügbar (akt. Marktpreis eine Ipv4-Adresse außerhalb von Afrika rund 15 €)
- bis 2022 sind geschätzt rund 50 Mrd. Adressen für IoT-Geräte notwendig
- praktisch wird Netzwerk-Design problematisch
- es muss mit NAT (Network Address Translation) bzw. CGNAT (Carrier Grade NAT) gearbeitet werden
- es kann im Fehlerfall nur eines Gerätes zu Mehrfachausfällen auch an anderen Geräten kommen (Fate Sharing)
- Fehler am Router betrifft häufig das gesamte Netz
- Security-Vorfälle betreffen häufig ganze (Teil-)Netze
- Adressen werden selbst in privaten Netzen knapp; Änderungen unerwünscht, weil aufwendig (auch Kosten-intensiv), Fehler-anfällig und Problem-behaftet
- ...

wegen Mangel an nutzbaren Adressen und der immer mehr steigenden Anzahl von Netzwerk-Geräten (vom Computer über Fernseher bis zum Rolladen usw. usf.) musste neue Adressierungs-Art eingeführt werden

neue Art sollte viele Jahrzehnte Bestand haben können, deshalb zweifache Skalierung:

Anzahl der Nummern-Blöcke von 4 auf 6 vergrößert

die Größe der Nummern in einem Block von 256 auf 65536 erhöht

Nummern werden als vier Hexadezimalziffern notiert und mit Doppelpunkt getrennt

bestimmte Teile mit ausschließlich Nullen können vereinfacht geschrieben werden, dadurch sind u.U. weniger Blöcke in Verwendung

### **Aufgaben:**

- 1. Berechnen Sie die theoretisch verfügbare Adressen-Anzahl für das IPv6-System!***
- 2. Schätzen Sie, wieviele Adressen pro Quadratmillimeter Erdoberfläche bei gleichmäßiger Verteilung (die Erde sei eine Kugel!) bereit stehen könnten!***
- 3. Berechnen Sie wieviele Adressen theoretisch für einen Quadratmillimeter Erdoberfläche zur Verfügung stehen (bei immer noch gleichmäßiger Verteilung!)***
- 4. Vergleichen Sie IPv4 und IPv6!***

128 bit lang; 8 Blöcke aus 4 Hexadezimalzahlen

2-Byte-Blöcke mit

Doppelpunkt getrennt

2001:0db8:0000:0000:03a4:0000:0000:9f21

führende Nullen dürfen

2001:db8:0:0:3a4:0:0:9f21

weggelassen werden

genau ein Block aus

2001:db8:0:0:3a4:0:0:9f21 → 2001:db8::3a4:0:0:9f21

(nur) Nullen kann durch

2001:db8:0:0:3a4:0:0:9f21 → 2001:db8:0:0:3a4::9f21

:: ersetzt werden

verkürzte Schreibweis nur für die Kommunikation mit dem menschen oder der Menschen untereinander gedacht, für Computer bleibt es eine 128-bit-Zahl (16-Byte-Zahl)

IPv6-Adresse besteht aus zwei gleichgroßen Teilen je 64 bit oder 8 Byte (bzw. 4 (Byte-)Word / 16-bit-Word)

die ersten 8 Byte sind der Network Identifier

Network Identifier Interface Identifier  
Host Identifier

die letzten 8 Byte sind der Interface Identifier

????:????:????:????:????:????:????:????

damit ist jedes Netzwerk (Ethernet-Segment unter IPv6)

2001:db8::3a4:0:0:9f21

immer ein /64-Netzwerk

2001:db8:0:0:3a4::9f21

es können ausnahmensweise auch kleinere Netzwerke festgelegt werden, dann entfallen aber die Host-Adresse mit den höheren Nummern

die ersten 64 bit sind der Netz-Präfix;

beginnt bei automatische Adress-Vergabe (beim Start eines Rechners) mit fe80; alle anderen Bits sind 0

der Internet-Provider vergibt dann später einen neuen Netz-Präfix für die Identifizierung im Internet

letzte 4 Blöcke (64 bit) sind Host-Teil

sie wird aus MAC-Adresse berechnet: in der MAC-Adresse wird im 1. Byte das 7. Bit invertiert; in diese Byte-Folge wird mittig fffe eingefügt

### Aufgaben:

1. **Ermitteln Sie die automatischen IPv6-Adressen zu folgenden MAC-Adressen:**

a)

b)

c)

d)

2. **Eine automatische erzeugte IPv6-Adresse ist nur für die Kommunikation in einem LAN zweckmäßig. Begründen Sie diese Aussage!**

praktisch ist das  $(IPv4)^4$  also  $IPv4 \times IPv4 \times IPv4 \times IPv4$

insgesamt jetzt  $2^{128} = 340'282'366'920'938'463'374'507'431'768'211'456$  Adressen verfügbar  
 $340 \times 10^{33}$  Adressen → 340 Pentrionen Adressen

Anzahl Adressen reicht aus, um für jeden Quadratmillimeter der Erdoberfläche  $665'570'793'348'866'944$  ( $665 \times 10^{15} = 655$  Trillionen) bereit zu stellen. Das sind mehr als heute mit IPv4 insgesamt vorhanden sind.

---

neue Parsung der Adressen notwendig (Umstellung von Software auf verschiedenen Schichtebenen)

### **Adress-Vergabe**

#### **IANA () vergibt Adress-Bereiche an (Regional Registrys):**

- LACNIC (Latin America and Caribbean NIC)
- APNIC (Asia Pacific NIC)
- RIPE (Réseaux IP Européens), speziell: RIPE NCC
- ARIN (America Registry for Internet Numbers)
- AfriNIC (Africa NIC)

diese vergeben dann wieder Unterbereich an die nationalen Registrationen

für den heimischen PC und die anderen Geräte im heimischen Netz werden die Adressen dann vom Provider vergeben (erfolgt hier automatisch)

für Firmen usw. besteht die Möglichkeit:

- von einem Provider eine / mehrere Adressen zu beziehen (erfolgt meist automatisch)
- direkter Kauf einer IPv6-Adresse mit einer Mitgliedschaft in der RIPE
- über den Provider als gesponsorter Vertreter

Internet ist in tausenden Autonomen System (AS)

jedes AS hat eine eindeutige Nummer

Vergabe der Nummern über die RIPE NCC ähnlich wie bei den IPv6-Adressen

über den Anschluss an einen, zwei oder mehrere Provider kann man eine eigenständige AS werden

man wird so Teil des Internets

auch Anschluss an Peering-Punkte ist für die Organisation eines AS

Grundprotokoll der AS untereinander ist BGP4 (Border Gateway Protocol 4)

stabiles und robustes IPV

---

## Netzwerk-Klassen (IPv6)

praktisch auch CDIR realisiert  
also Klassenlose Adressierung, wegen der schon so vorhandenen Adressen-Anzahl wenig genutzt

alle Netze sind pro forma /64-Netzwerke  
werden kleinere Netzwerke gebraucht, dann lassen sich diese zwar festlegen, die höheren Netzwerke sind dann nicht nutzbar

Maskierung würde jetzt über F's erfolgen, da diese in der Hexadezimal-Darstellung für die volle Besetzung der Biststellen mit Einsen entspricht  $F_{16} = 1111_2$

**IPv6-Adresse:**            **????:????:????:????:????:????:????:????**  
**Netzwerk-Maske (/64):**    **FFFF:FFFF:FFFF:FFFF:0000:0000:0000:0000**

IPv6 ist auch heute (2018) noch in der Frühphase der Umsetzung (vor allem durch den Parallel-Betrieb von IPv4 und IPv6 bedingt)

in einem /32-Netzwerk beginnt die IPv6-Adresse mit dem von der RIPE festgelegten Network Identifier gefolgt von 32 bit für die eigene Vergabe  
d.h. praktisch kann jeder IPv6-Adress-Besitzer in einem /32-Netzwerk rund 4,3 Mrd. eigene Netzwerke anlegen; jedes dieser Netzwerke könnte rund 18,5 Mrd. Endpunkte (Hosts) beinhalten

**IPv6-Adresse:**            **RIPE-            eigene            Hosts in einem**  
                                 **Vorgabe        Netze            eigenen Netz**  
**Netzwerk-Maske (/64):**    **????:????:????:????:????:????:????:????**  
                                 **FFFF:FFFF:0000:0000:0000:0000:0000:0000**

meist ein Standard-Gateway aus einem anderen Adress-Bereich



---

## Arten von IPv6

- **Link Local Adress** FE80::/10
- **Site Local Adress** FEC0::/10  
als Ersatz für lokale Netzwerke gedacht (also wie Klasse C, B- und A-Netzen in IPv4; z.B.: 192.168.0.0/16)  
derzeit abgeschafft; !nicht benutzen; im Router Weiterleitung verhindern (in beiden Richtungen)!
- **Unique Local Adress** FC00::/7  
unterteilt in zwei Unterbereiche:
  - FD00::/8 → frei nutzbar
  - FC00::/8 → für eine ev. zentrale Registrierung reserviertnur lokale Benutzung zugelassen (auch bei den offiziell vergebenen Adresse auch dem FD00-Bereich müssen ebenfalls im Router blockiert werden  
nur lokales Reverse DNS möglich (globales RDNS geht nicht!)
- **Documentation Prefix** 2001:DB8::/32  
Netzwerk nur für unspezifische / allgemeine Dokumentationen  
kein echtes nutzbares Netz!!! darf praktisch nicht verwendet werden  
in Firewall eingehend u. ausgehend verwerfen / rausfiltern
- **Global Unicast Adress** 2000::/3  
öffentliche Internet-Adressen  
(derzeit nur 1/8 des verfügbaren Adress-Raumes benutzt, der Rest wird derzeit noch nicht vergeben (riesige Reserve))  
nur dieser Raum sollte in der Firewall gestattet werden  
reverse DNS ist vorhanden und nutzbar  
Vergabe der Adressen erfolgt über die RIR (in Europa ist es die RIPE)
- **Multicast Adress** FF00::/8  
zu erkennen an den 2 F's am Anfang  
praktische Anwendungen:
  - Verteilung von Inhalten
    - Video
    - Börsenkurse
  - Automatisierung im Netzwerk
    -im Normalfall in der Firewall verwerfen (nur bei speziellen Anwendungen durchlassen)  
der Adressbereich FF02::/16 ist für Multicast Link Local vorgesehen  
(in Verbindung mit den Link Local Adressen (FE80::/10) können sich Rechner und Router usw. automatisch im Netzwerk finden; Aktivierung sofort mit dem Interface; sofort Kommunikation auf Layer 3 möglich)

---

Gemeinsamkeiten aller IPv6-Adressen:  
128 bit lang  
64 bit Network Identifier  
64 bit Interface Identifier / Host Identifier

### **"Global Unicast"-Adressen**

öffentliche Adressen für den Verbindungsaufbau ins Internet

### **"Link Local"-Adressen**

beginnen mit FE80:: ...  
in Windows 10 "Verbindungs**lokale** IPv6-Adresse" genannt (vielleicht etwas ungünstig eingedeutscht)

eine Zweite Link Local Adresse ist der Standard-Gateway

	<b>normal</b>	<b>Host-Identifizier</b>	<b>Windows-</b>
	<b>Vorgabe Nullen</b>	<b>(zufällig)</b>	<b>Interface</b>
<b>IPv6-Adresse:</b>	<b>FE80:0000:0000:0000</b>	<b>????:????:????:????</b>	<b>%16</b>
<b>verkürzt:</b>		<b>FE80::????:????:????:????</b>	<b>%16</b>

jedes Interface bekommt sofort und immer eine Link Local Adresse und kann damit auf Layer 3 (→ ISO-OSI-Modell) arbeiten  
deshalb kein Warten z.B. auf einen DHCP-Server (wie bei IPv4) und dann erst Kommunikation über Layer 2  
Pakete an Link Local-Adressen werden nicht ins Internet geroutet  
sie sind im eigenen Netz (LAN) eingesperrt

### **Netzwerk-Automatisierung**

neues Feature von IPv6 (bei IPv4 nicht vorhanden)  
ermöglicht das gegenseitige Finden von Device's im Netzwerk (Drucker, Rechner, Router, Server, ...)  
automatisches Finden / automatisches Suchen des oder der Router über Router Solicitation (RS; ICMPv6 Paket)  
aktives Melden des oder der Router über Router Advertisement (RA; ICMPv6 Paket) (auf Anfrage durch Router Solicitation); auch periodisch Absendung

ICMPv6-Paket ist ein Layer 3 Paket, welches Link Local und Link Local Multicast-Adressen beinhaltet  
im vom Router zurückgesendeten Paket befindet die Information lokalen Global Unicast Netzwerk (den im Netzwerk gültigen Network Identifier)) und auch wirklich nur diesen Teil einer IPv6-Adresse

---

Network Identifier Interface Identifier  
Host Identifier  
2001:0db8:0000:0000:03a4:0000:0000:9f21  
2001:db8:0:0::/64

der Documentation Prefix 2001:db8::/64 muss im konkreten Netz durch die (zugewiesene) Netzwerk-Adresse ersetzt werden  
der Host Identifier muss der Host selbst festlegen  
damit völlig automatisch Adress-Festlegung (kein administrativer Eingriff mehr notwendig)  
aber trotzdem statische Konfiguration möglich (z.B. Server, Drucker, ...)  
Default-Gateway wird am Rechner / Host automatisch aus der Absende-Adresse des Router Advertisement Paketes entnommen und beim Host eingetragen  
die Information über den lokalen DNS-Server kann über das Router Advertisement Paket erfolgen oder man nutzt einen DHCPv6-Server

## **DHCPv6**

unterschiedliche Funktionen und Leistungen im Vergleich zu IPv4  
DHCPv6 regelt nicht die lokale Adress.Vergabe (die ist schon vorher automatisch erfolgt)  
Übertragung beschränkt sich auf zusätzliche Informationen (über Router Advertisement: z.B. DNS, Time-Server, SIP-Server, ...)  
da der DHCPv6-Server keine Tabellen mehr führen muss, spricht man vom stateless DHCPv6  
für Linux und Windows 10 (ab Version 1703) verfügbar  
nicht bei Android (weil der Mobilfunk das derzeit nicht unterstützt; Wechsel von Funkzellen bereit technisches Problem)

beim stateful DHCPv6 fragt Rechner nach einer Adresse und den zusätzlichen Information (wie beim stateless DHCPv6)  
Rechner bekommt zusätzlich eine IPv6-Adresse zurück, die beim Server in einer Tabelle eingetragen wird  
diese neue Adresse bekommt der Rechner zusätzlich zu seinen – vorher schon – generierten Adressen  
so kann es passiern, das auf einem Interface bis zu drei Adressen vergeben sind (alle sind nutzbar!)  
für eingehende Verbindungen alle nutzbar, für ausgehende Verbindungen wird nur die temporäre Adresse (Global Unicast) genutzt (nicht die vom DHCP-Server)  
DHCPv6-Server liefert nicht die Default Gateway Adresse (die kommt vom Router über RA)

	<b>DNCPv4</b>	<b>DHCPv6</b>
<b>Gemeinsamkeiten</b>	stateful DHCP möglich (aber unterschiedliche Funktionabilität!)	
<b>Unterschiede</b>	nur stateful DHCP Konfiguration des Interface bestimmt über DHCP Vergabe einer IPv4-Adresse  nur eine Adresse aktiv Übermittlung der Netzwerk-Maske und Netzwerkmaske (damit auch Info über Netzwerkgröße)  Vergabe des Default Gateway	stateful und stateless DHCP Steuerung erfolgt über Router Advertisement Vergabe einer zusätzlichen IPv6-Adresse es sind mehrere Adressen aktiv keine Übermittlung der netzwerkmaske oder informationen zur Netzwerkgröße keine Information zum Default Gateway (läuft über Router Advertisement) liefert zusätzliche Adressen (z.B. DNS, Time-Server, SIP-Server, ...) ...

### **IPv6 unter microsoft® Windows® 10**

in Update-Version 1607 gibt es einen ungefixten Fehler bei der IPv6-Verarbeitung

über die Kommandozeile `cmd` lassen sich mit dem Befehl `ipconfig` die Konfigurationen der aktuellen Netze anzeigen; ev. kann man die Ausgaben noch über die Option `/a11` hinter `ipconfig` (Lehrzeichen-getrennt!) erweitern

bei IPv6 sind einem Interface fast immer mehrere Adressen zugeordnet (bei IPv4 war das praktisch immer nur eine)

Adress-Vergabe erfolgt automatisch und ohne Einwirken von Nutzer oder Adminsitrator  
 System hat mehrere Möglichkeiten eine Adresse festzulegen

alle modernen Betriebssystem unterstützen IPv6  
 Kommunikation wird sofort aktiv

- suchen Nachbarn
- suchen den Router
- tauschen Daten aus

praktisch wird IPv6 immer aktiv; eine einfache Deaktivierung ist nur oberflächlich  
 Geräte kommunizieren von sich aus auf IPv6-Ebene  
 es ist eher sinnvoll IPv6 aktiv zu nutzen, als es zu "deaktivieren" (und dadurch unbeobachtet wirkt)

---

## Internet-Ressourcen-Adressen

Uniform Resource Locator (URL)

Nutzung von Ressourcen im Web-Browser

Ressourcentyp://Nutzer:Passwort@Computer.Domäne.TopLevelDomäne:Port/Pfad/Datei  
Parameter

### **Bestandteile einer URL**

- **Ressourcentyp**                    verwendete Internet-Protokoll  
                                         z.B. http, ftp, ...
- **:**                                    trennt Ressourcen-Typ und Ressourcen-Zeiger voneinander
- **Doppelpunkt**
- **/**                                    // ... nicht lokale Ressource → Aktion auf entferntem Rechner  
                                         /// ... lokale Ressource → Aktion auf dem eigenen Rechner  
                                         (dritter Schrägstrich steht hier für die oberste Ebene (Root-Verzeichnis),  
                                         angelehnt an UNIX/Linux-Systeme)
- **Schrägstriche / Slash**
- **Nutzer**                            optional
- **Passwort**                        optional
- **@**                                    Ende des Nutzer-Passwort-Bereiches, kann weggelassen  
                                         werden, wenn keine Nutzer-Kennung verwendet wird
- **at-Zeichen /**  
  **Klammeraffe**
- **Computer**                        genaue Adresierung von Rechner, Domäne und Land /  
  **Host**                                Toplevel-Domäne
- **Domäne**                            kann insgesamt durch eine IP-Adresse ersetzt werden
- **Top Level Domäne**
- **Port**                                besonderer Port für TCP- bzw. UDP-Services
- **Pfad**                                genaue Angabe / Lage einer Ressource / Datei
- **Datei**                                hier ist unbedingt auf die exakte Notierung von Groß- und  
                                         Klein-Buchstaben zu achten!
- **Parameter**                        Beeinflussung der Ressource / Start-Parameter

### **Beispiele:**

file:///C:/Windows/

http://www.web.de

ftp://ftp.beispiel.de/daten/beispiel.zip

mailto:beispiel@web.de

http://www.beispiel.com/

http://www.beispiel.com/index.shtml?variable=wert

http://www.beispiel.com/index.html#impressum

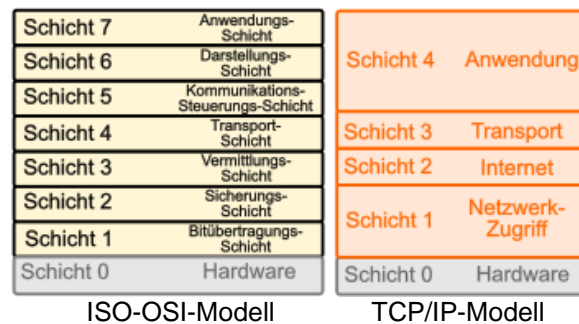
---

## OSI-Adressen

zusammengesetzt aus bis zu 20 Oktetts  
so konzipiert, das eine OSI-Adresse weltweit einmalig ist  
Ort-unabhängig, da auf den Computer / Host mit seinen Anwendungen spezifiziert  
besteht aus Initial Domain Part (IDP) und dem Domain Specific Part (DSP)

der IDP ist für verschiedene Adress-Domänen standardisiert  
enthält Länder-Merkmal  
innerhalb der DSP gilt die Adressierungs-Autorität für die spezielle Domäne  
hier kann der Subnetz-Betreiber eigenständig Festlegungen treffen  
meist werden Knoten-Adressen mit benutzt (dies hebt aber i.A, die Orts-unabhängigkeit wieder aus, was nicht gewünscht ist)

OSI-Netzwerk-Adresse + bis zu drei Sektoren für jeweils die Transport-, Sitzungs- und Darstellungs-Schicht (OSI-Schichten 4 - 6)  
Schicht 4 gehört zur Transport-Schicht im TCP/IP-Modell, die anderen beiden Schichten werden hier der Anwendungs-Schicht zugeordnet



OSI-Adressen werden innerhalb der Darstellungs-Schicht (OSI-Schicht 6) verwendet  
sie dient der Adressierung von Anwendungs-Instanzen

Zerlegung nur über spezielle Tabellen möglich, die bei den IMP's hinterlegt sind

## NIC-Adresse

Network Interface Card Adresse → MAC-Adresse

---

## Transport-Schicht

Transmission Control Protocol (TCP)

Verbindungs-orientiert  
Fluss-Kontrolle  
zuverlässig

HTTP	IMAP	SMTP	...	DNS	...
TCP				UDP	
IPv4				...	
ARP (Address Resolution Protocol)					
Ethernet		Token-Ring		...	

User Datagramm Protocol (UDP)

Verbindungs-loses Protokoll  
unzuverlässig

HTTP	IMAP	SMTP	...	DNS	...
TCP				UDP	
IPv4				...	
ARP (Address Resolution Protocol)					
Ethernet		Token-Ring		...	

## ASCII-Code-Tabelle (Basis-Version)

HEX	MSD	0	1	2	3	4	5	6	7
LSD	bits	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SPACE	0	@	P	-	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HAT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	--
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	O	^	n	~
F	1111	SI	US	/	?	N	←	o	DEL

### wichtige Codes

NUL	NULL	leeres Datenfeld / Byte	
SOH	Start of Heading		
STX	Start of Text		
ETX	End of Text		
EOT	End of Transmission		
ENQ	Enquiry		
ACK	Acknowledged		
BEL	bell	Signalausgabe	
BS	Backspace		[ ← ]
HT	Horizontal Tabulation	Horizontaler Tabulator	[ Tab ]
LF	Line Feed	Zeilenvorschub	
VT	Vertical Tabulation	vertikaler Tabulator (Seitenvorschub)	
CR	Carriage Return	Wagen Rückfahrt / Zeilenrücksprung	[Enter] → LF + CR
SO	Shift Out		
SI	Shift IN		
DLE	Data Link Escape		
DC	Device Control		
NAK	Negative Acknowledge		
SYN	Synchronous Idle		
ETB	End of Transmission Block		
CAN	Cancel	Abbruch	
EM	End of Medium	Papierende	
SUB	Substitute		
ESC	Escape		[Esc]
FS	File Separator		
GS	Group Separator		
RS	Record Separator		
US	Unit Separator		
SPACE / SP	Space / Blank	Leerzeichen	[ ]
DEL	Delete	Löschen	



---

### Binär-coded-Decimal - BCD

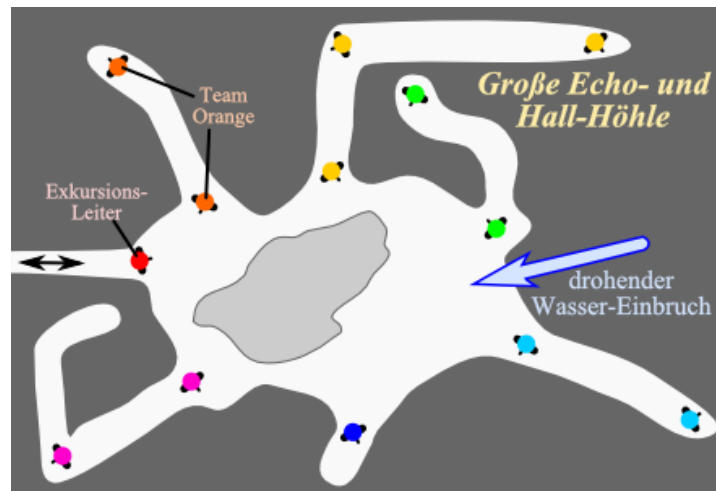
dezimale Ziffer	BCD		...	Hunderter	Zehner	Einer
0	0000		...	$10^2$	$10^1$	$10^0$
1	0001			0	0	0
2	0010			1	1	1
3	0011			2	2	2
4	0100			...	...	...
5	0101			9	9	9
6	0110					
7	0111		z.B.:	3	7	5
8	1000			0011	0111	0101
9	1001					

---

### 3. Protokolle - Grundlagen

#### Aufgaben:

1. *Teilen Sie sich im Kurs in kleine Gruppen von maximal 4 Teilnehmern! Zwei gehen vor die Tür, zwei bleiben im Raum. Das eine Teil-Team bekommt vom Kurs-Leiter eine spezielle Aufgabe.*
2. *Mit Hilfe von 3 Seilen mit einer Länge von mindestens 5 m bauen Sie auf dem Flur oder dem Schulhof eine Stern-Struktur auf (einer in der Mitte hält die drei Seile, die anderen ordnen sich maximal entfernt an. Nun lösen Sie die Aufgaben auf den Kärtchen, die der Kurs-Leiter verteilt hat! Merken Sie sich die auftretenden Probleme!*
3. *Wiederholen Sie die Aufgabe 2! Dieses Mal dürfen Sie sich vorher absprechen. Notieren Sie kurz die Absprachen!*



#### Aufgaben:

1. *Sie befinden sich mit einer Gruppe anderer Höhlenforscher in einer riesigen, hallig klingenden Höhle mit vielen Seitenarmen. Jeder Forscher hat einen Seitenarm erforscht und wartet jetzt am Eingang seines Seitenarmes, um den anderen die Erkenntnisse über mögliche Verzweigungen und Ausgänge mitzuteilen. Die Zeit ist begrenzt, da es durch Regen außerhalb der Höhle zu vermehrtem Wassereinbrüchen in das Höhlensystem kommt. Entwickeln Sie ein Protokoll, damit jeder seine Informationen an die anderen weitergeben kann!*

Was braucht man für eine funktionierende indirekte Kommunikation?

---

## Was sind Protokolle?

sehr alter Begriff

kommt vom mittelgriechischem protokollon (dt.: Leim, Klebe) und dann folgend ausmittellateinischen protocollum (für: amtliche Papyrus-Rolle; Verhandlungsbericht) ins Deutsche übernommen wurde

auch als Begriff für "Sammlung von Regeln" in Mittelalter genutzt

<b>Definition(en): Protokolle</b>
Protokolle sind Verfahrens-Vorschriften bzw. -Vereinbarungen, die Zeitpunkte, Reihenfolgen und Inhalte (Inhalts-Typen u. -Mengen) von Kommunikationen bestimmen.
Ein Protokoll ist eine Regelwerk, das den Vorgang der Daten-Übertragung / Kommunikation zwischen zwei Kommunikanten beschreibt.

---

### ***Protokolle in unserem Leben, ... (und darüber hinaus)***

- **Geburts-Urkunde**
- **Krankschreibung**
- **Kontoauszug, Sparbuch** Protokolle über unseren Umgang mit Geld
- **Totenschein**
- **Umgangsformen** Begrüßung, Anstellen an der Kasse, ...
- **Tagesordnung** Ablauf-Vereinbarung für Versammlungen, ...
- **Wohnungsübergabe-Protokoll** Auflistung von Mängeln usw. bei einer Wohnungs-Übergabe
- **Unfall-Protokoll** Niederschrift zu einem Unfall-Hergang
- **Stenogramm** Schnell-Mitschrift von Reden, diktierten briefen usw. (z.B. Bundestag, Sekretariaten, ...)
- **SMS** Kurznachrichten-Protokoll (Short Message Service) auf einem Handy / Smartfon
- **WhatsApp** moderner Instant-Message-Dienst auf Smartfons und Tablets
- **Prüfungs-Protokoll** Niederschrift über den verlauf und das Ergebnis einer Prüfung
- **diplomatisches Protokoll** roter Teppich, Kleiderordnung, Tischordnung; Verbeugungen, ...
- **Log-Dateien** Berichte zu Kommunikationen und Installationen auf Computern
- **Start-Protokoll für ein Raumschiff**
- 

### **Aufgaben:**

- 1.
- 2.
- 3.

---

## **3.x. Kommunikations-Modi**

### **3.x.1. synchrone Datenübertragung**

Sender und Empfänger kommunizieren gemeinsam / wechselseitig / abhängig voneinander

Warten der Kommunikanten aufeinander notwendig

#### ***Beispiele für synchrone Datenübertragungen***

- **Telefonie**
- **Terminal-Server-Sitzungen**
- **Videokonferenzen**
- **ISDN**
- 

### **3.x.2. asynchrone Datenübertragung**

Sender und Empfänger kommunizieren losgelöst voneinander

Puffer-Mechanismen (Zwischen-Speicher) notwendig

#### ***Beispiele für asynchrone Datenübertragungen***

- **SMS**
- **eMail**
- **Instant Messaging**
- **ADSL**
- 

#### ***Vermittlungs-Arten***

- **Leitungs-Vermittlung**
- **Paket-Vermittlung**
-

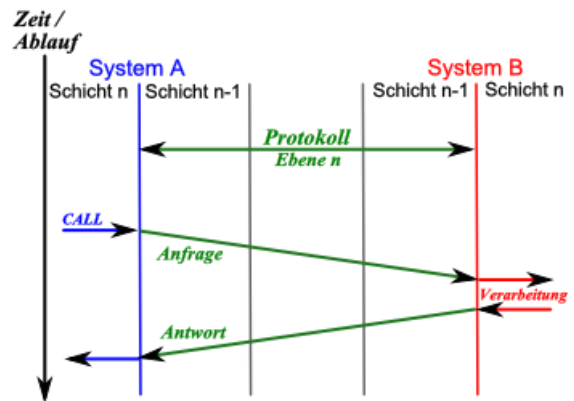
# Verbindungs-orientierte Daten-Übertragung

## **Ablauf**

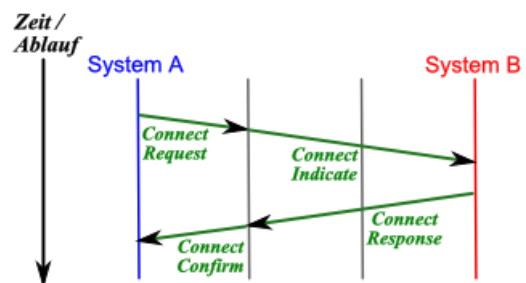
- 1. Aufbauphase für Verbindung**      Verbindung wird aufgebaut
- 2. Verbindungsphase**              Verbindung wird zur Daten-Übertragung genutzt
- 3. Abbauphase für Verbindung**      Verbindung wird beendet

alle Phase im ständigen Handshake-Verfahren; erst wenn eine Phase abgeschlossen und dies (rück-)bestätigt ist, wird mit nächster Phase fortgesetzt

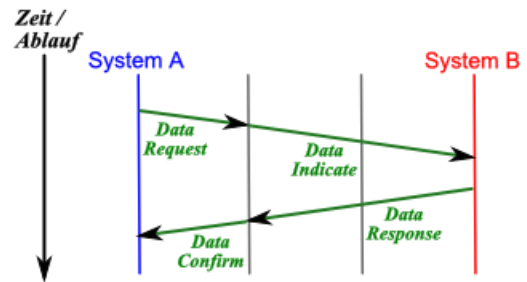
ständiger Wechsel zwischen Anfrage und Antwort und ev. Warten / Lauschen



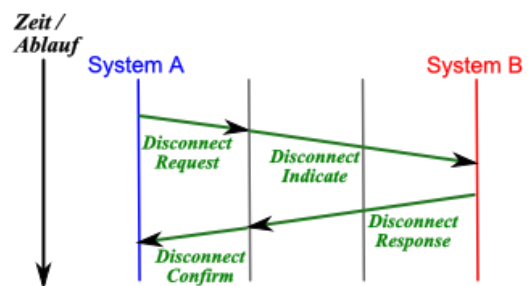
Grundbegriffe / allgemeines Schema einer Kommunikation über eine bestimmte Protokollebene



Ablauf des Verbindungs-Aufbaus



Ablauf der Datenübertragung  
(wird solange wiederholt,  
bis alle Daten übertragen wurden)



Ablauf des Verbindungs-Abbaus

### Definition(en): Verbindungs-orientierte Kommunikation

Eine Verbindungs-orientierte Datenübertragung ist eine wechselseitige Kommunikation, deren Ablauf von der vorlaufenden Kommunikation abhängt.

Eine Verbindungs-orientierte Datenübertragung ist eine wechselseitige Kommunikation, bei der jeweils der (temporäre) Sender auf die Antwort (Quittierung oder Datenlieferung) durch den (temporären) Empfänger wartet und dann die Kommunikation (ev. wechselseitig) gleichartig fortsetzt wird.

Eine Verbindungs-orientierte Datenübertragung ist eine wechselseitige Kommunikation, nach einem sich gegenseitig bedingenden Anfrage-Antwort-Protokoll-System.

TCP (Transmission Control Protocol)

zwischen zwei Endpunkten (Sockets) wird eine wechselseitige Kommunikations-Verbindung hergestellt

gehört zur Transport-Schicht (Schicht 4 (von 7)) des ISO-OSI-Modells  
im DoD-Modell ist die Host-Schicht (Schicht 3 (von 4))  
Adress-Vergabe (IP-Protokoll) jeweils eine Schicht tiefer

---

### **Verbindungs-orientierte Internet-Protokolle**

- **HTTP** Hypertext Transport Protocol
- **SMTP** Post Office Protocol
- **POP** Post Office Protocol
- **IMAP** Internet Message Access Protocol
- **FTP** File Transport Protocol
- **Telnet** Teletype Network

Simplex

Halb-Duplex

Duplex  
Voll-Duplex



---

## Verbindungs-lose Daten-Übertragung

einzelne Schritte sind nicht mehr zeitlich so streng aneinander gereiht, Quittierungen entfallen teilweise und

Maximale Verbindungs(aufbau)zeit festgelegt, damit Kanäle nicht unendlich belegt (von Irrläufern) oder benutzt werden

### **Definition(en): Verbindungs-lose Kommunikation**

Eine Verbindungs-lose Datenübertragung ist eine vorrangig einseitige Kommunikation, bei der die Übertragung durch den Sender unabhängig vom Empfänger vorgenommen wird.

Broadcast ist eine typische Verbindungs-lose Kommunikation

UDP (User Datagramm Protocol)

### **Verbindungs-lose Internet-Protokolle**

- **DNS**      Domain Name Service
- **DHCP**     Dynamic Host Configuration Protocol
- **NFS**      Network File System
- 
- 

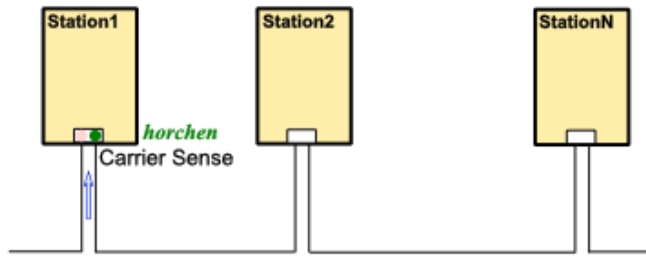
theoretisch 65535 Ports möglich, praktisch in den Systemn nur die Ports bis 1024 oder 4096 umgesetzt

# CSMA/CD-Zugriffs-Verfahren

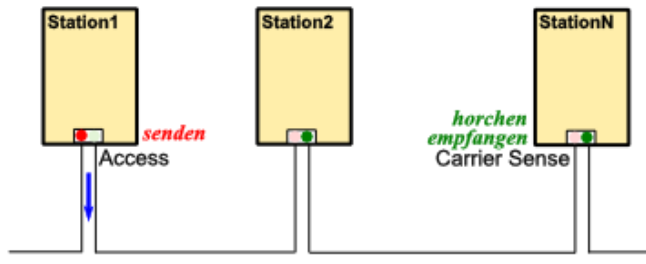
## Carrier Sense Multiple Access / Collision Detection

nur senden wenn kein anderer sendet (senden, wenn das Medium frei ist)

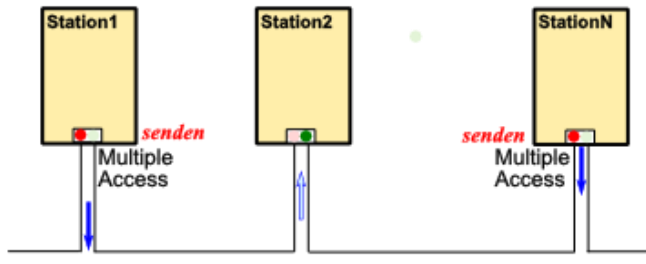
eine Station, die neu in Netzwerk intritt, horcht zuerst einmal, ob auf dem Medium / Bus schon ein Signal unterwegs ist (Carrier Sense)



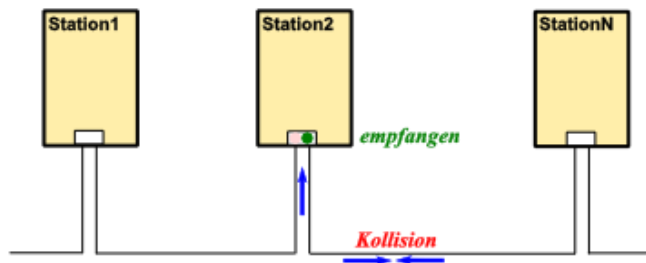
ist das Medium frei, dann kann jetzt ein Senden / der Zugriff (Access) erfolgen



sendet zur gleichen Zeit eine andere Station ebenfalls, dann treffen sich die Signale zwischen den beiden Station, es kommt zur Signal-Verfälschung



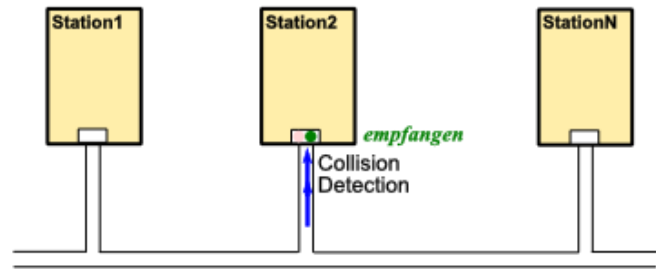
die Signale kollidieren  
Stationen erkennen dies



---

Signale überlagern sich und sind damit nicht exakt empfangbar

nach einer zufälligen Ruhepause wird wieder versucht (nach dem Horchen) ein Signal zu senden



Verfahren ist nicht echtzeitfähig, da die Wartezeiten nicht voraussehbar sind und welche Station wann einen Sendezugriff bekommt

bei großen Zahlen an sendewilligen Stationen nimmt Effizienz des Verfahrens ab  
ab rund 40% Auslastung des Mediums nimmt die Anzahl von Kollisionen stark zu und nimmt die Gesamtverarbeitungs-Geschwindigkeit ab (steigt nur noch schwach)

**Aufgaben:**

***1. Beschreiben Sie den Ablauf des CSMA/CD-Verfahrens!***

***2.***

***3.***

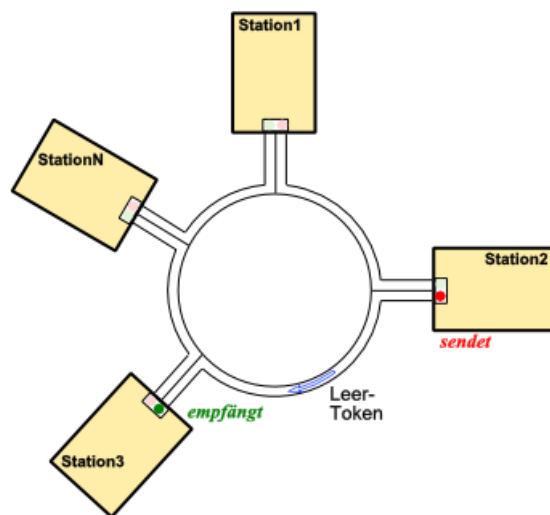
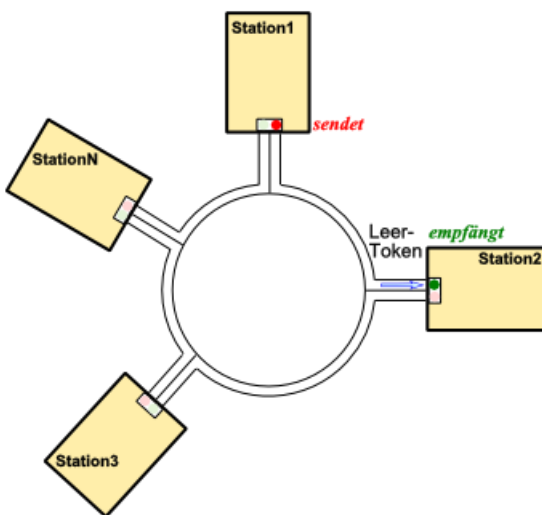
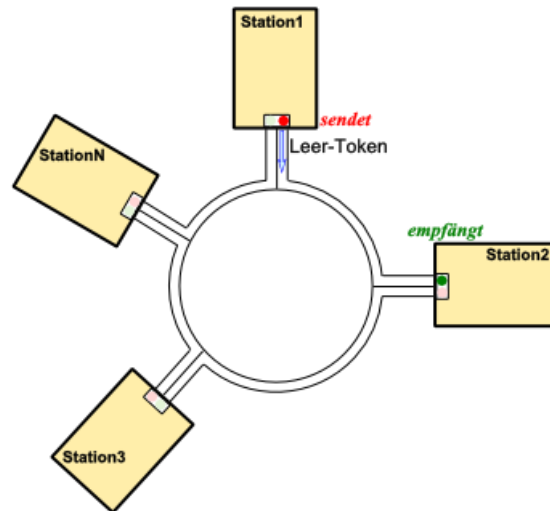
***!***

## Token-Ring-Verfahren

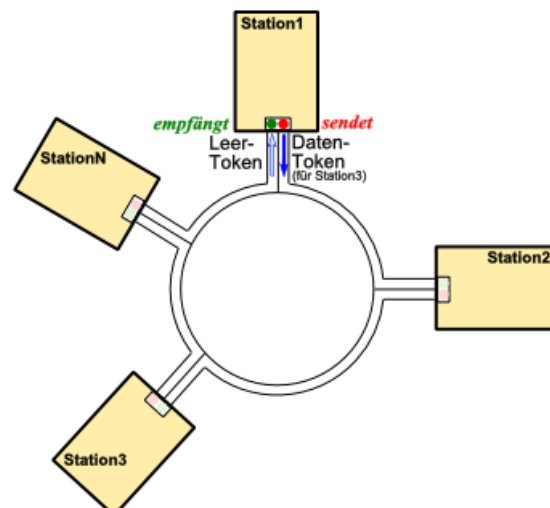
Ring besteht aus eindeutigen Punkt-zu-Punkt-Verbindungen

es wird ständig ein Arbeits-Zeichen (Token) von Station zu Station im Ring weitergereicht

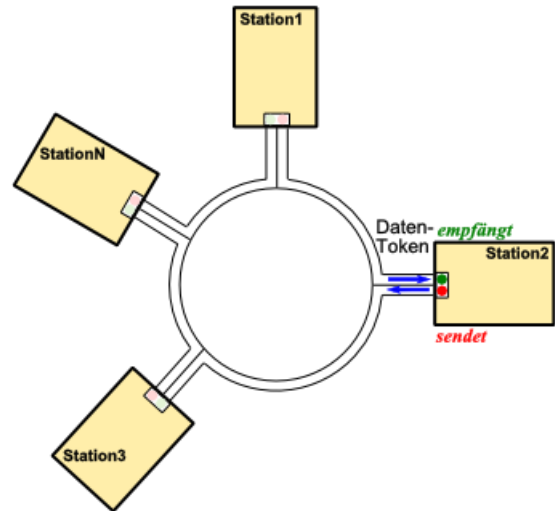
die Station, die den Token hat, darf senden  
gibt es nichts zu senden, dann wird ein sogenannter Leer-Token zur nächsten Station gesendet



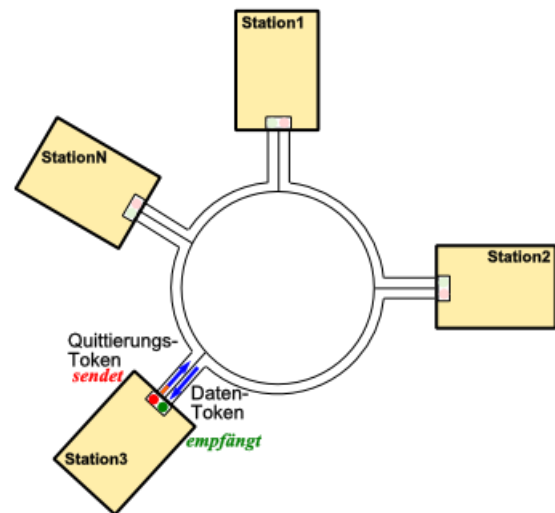
sendewillige Station, die den Token hat  
hängt die Daten an den Token an



nachfolgende Stationen prüfen, ob sie der Empfänger sind



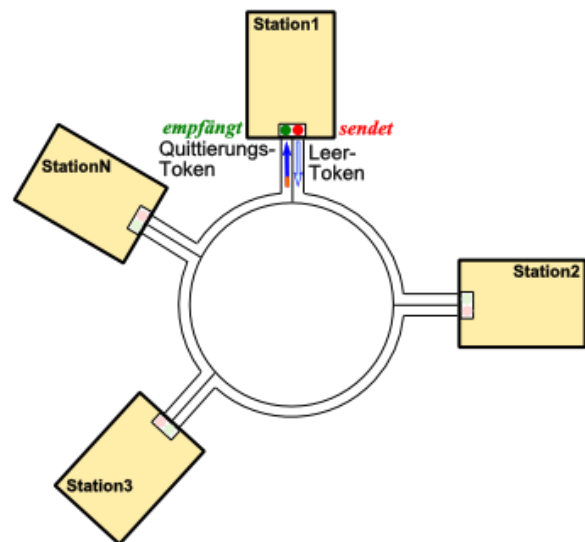
wenn ja, dann werden Daten kopiert und ein Quittierungs-Flag gesetzt  
der Token wird dann wieder weitergegeben



Sende-Station erhält Token mit Quittierung  
sendet entweder neue Daten oder einen Leer-Token

insgesamt darf eine Station aber nur eine bestimmte Zeit den Token für sich beanspruchen

damit wird eine Netz-Gerechtigkeit garantiert und alle Stationen können mit einem kalkulierbaren Warte-Zeit auch auf das Netz zugreifen



eine Monitoring-Station überwacht den Daten-Verkehr und erkennt Fehler

---

z.B. wenn die Sender-Station ihren quittierten Daten-Token nicht vom Ring nimmt  
wenn ein quittierter Daten-Token ein zweites Mal die Monitor-Station erreicht, dann entfernt sie diesen und ersetzt ihn durch einen Leer-Token  
auch wenn nach einer bestimmten Zeit kein Token mehr vorbeikommt, dann erzeugt die Monitor-Station einen neuen Leer-Token

arbeitet mit 4 Mbit/s ist aber das verwendete Verfahren ungefähr so schnell, wie 100 Mbit/s-Ethernet

in neueren Versionen (16 Mbit/s) kreisen mehrere Token im Ring

HTTP	IMAP	SMTP	...	DNS	...
TCP				UDP	
IPv4					...
ARP (Address Resolution Protocol)					
Ethernet		Token-Ring		...	

---

## ALOHA-Verfahren

immer senden, in der Hoffnung dass es klappt

Name nach der Begrüßung auf Hawaii-Inseln, da hier das Verfahren zum ersten Mal eingesetzt (1971, ALOHAnet)  
verband die diversen Einrichtungen der Universität von Honolulu auf den verschiedenen Inseln

es gibt das unsynchronisierte ALOHA-Verfahren und das synchronisierte.  
Beim unsynchronisierten Verfahren kann jede Station zu jeder Zeit senden.  
Das synchronisierte Verfahren beruht auf Zeit-Scheiben. Es gibt definierte – für alle Stationen synchron verlaufende – Zeit-Bereiche (slots), in den gesendet werden kann  
ein slot ist so lang, wie die festgelegte Daten-Paket-Länge des Verfahrens

wenn Daten kollidieren, dann sind Daten verstümmelt und werden nicht bestätigt

bei ausbleibender Quittierung werden die Daten nach einer zufällig bestimmten Zeit wieder gesendet

Daten-Durchsatz rund 18% der Kanal-Kapazität  
nicht echtzeit-fähig, da nicht sichergestellt werden kann, das Daten innerhalb einer bestimmten Zeit den Empfänger erreichen

Verfahren bildete die Grundlage des (vorne besprochenen) CSMA/CD-Verfahrens (→ ), welches sich wegen dem besseren Daten-Durchsatz durchgesetzt hat

Normen des OSI-Schicht-Modells

→ KALDEALI → S. 25

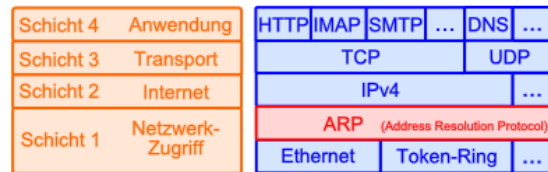
Anwendung OSI bei ISDN

# Netzwerk-Protokolle

## Address Resolution Protocol - ARP

ist praktisch die Schicht / das Protokoll, welche(s) die Kommunikation zwischen MAC-Adressen und den IP-Adressen.

das ist dann notwendig, wenn über das lokale Netzwerk hinaus kommuniziert werden soll die MAC-Adressen sind innerhalb eines Netzes bekannt



arbeitet im DoD-Modell (noch) in der Netz-Zugangs-Schicht

innerhalb des ISO-OSI-Modell's würde das ARP der Schicht 2 (Sicherheit) zugeordnet werden

Will ein Rechner im Netzwerk kommunizieren, dann schaut er zuerst in seinem eigenen / lokalen **ARP-Cache** (ARP-Tabelle, Routing-Tabelle) nach, ob schon ein gültiger Eintrag (aus IP- und MAC-Adresse) existiert. Das wird nach dem Rechner-Start nicht der Fall sein. Also versendet er zuerst einen sogenannten **ARP-Request** (ARP-Anfrage; Operation: 1). In diesem Paket befinden sich neben der eigenen MAC- und IP-Adresse auch die IP-Ziel-Adresse. Da die Ziel-MAC-Adresse ja nicht bekannt ist, wird die MAC-Broadcast-Adresse FF:FF:FF:FF:FF:FF verwendet. Somit geht die Anfrage an alle Netzwerk-Komponenten (mit einer MAC-Adresse) im Netz.

Byte	+1	+2	+3	+4
0	Hardware-Adress-Typ		Protokoll-Adress-Typ	
4	Hw-Adress-Größe	Prot.-Adress-Größe	Operation	
8	Quell-MAC-Adresse			
12	Quell-MAC-Adresse		Quell-IP-Adresse	
16	Quell-IP-Adresse		Ziel-MAC-Adresse	
20	Ziel-MAC-Adresse			
24	Ziel-IP-Adresse			

Jeder Rechner, der die Anfrage erhält – also am Netz hängt – prüft nun, ob er selbst die Ziel-IP-Adresse hat. Ist das der Fall, schickt er ein **ARP-Reply** (ARP-Antwort; Operation: 2) an den anfragenden Rechner zurück. Dessen Adressen hat er aus dem ARP-Request entnommen. Der anfragende Rechner kann nun aus dem ARP-Reply die IP-MAC-Adressen-Kombination entnehmen und in seinem ARP-Cache eintragen

Bei einem erneuten Kommunikations-Versuch, wird nun wieder in der ARP-Tabelle nachgeschlagen. Jetzt existiert ja ein gültiger Eintrag. Dieser wird nun Erstellung der IP-Pakete benutzt.

I.A. werden die Einträge schon nach einigen Minuten ungültig und es wird ein neuer ARP-Request notwendig. Das ermöglicht sehr dynamische Netze.



Anzeige des ARP-Cache mit arp (UNIX / Linux) bzw. arp -a (Windows)

Man erhält die Umsetztabelle für jede der verfügbaren Netzwerkschnittstellen.

Hier ist die 2. Schnittstelle eine virtuelle. Sie gehört zu einem VirtualBox-Netzwerk.

```

C:\Windows\system32\cmd.exe
C:\Users\drews>arp -a

Schnittstelle: 192.168.100.141 --- 0xc
Internetadresse    Physische Adresse    Typ
192.168.100.2      c8-0e-14-62-1e-82    dynamisch
192.168.100.105    30-cd-a7-11-c5-c4    dynamisch
192.168.100.159    fc-aa-14-70-8f-66    dynamisch
192.168.100.255    ff-ff-ff-ff-ff-ff    statisch
224.0.0.22         01-00-5e-00-00-16    statisch
224.0.0.252        01-00-5e-00-00-fc    statisch
239.255.255.250    01-00-5e-7f-ff-fa    statisch
255.255.255.255    ff-ff-ff-ff-ff-ff    statisch

Schnittstelle: 192.168.150.1 --- 0xe
Internetadresse    Physische Adresse    Typ
192.168.150.255    ff-ff-ff-ff-ff-ff    statisch
224.0.0.22         01-00-5e-00-00-16    statisch
224.0.0.252        01-00-5e-00-00-fc    statisch
239.255.255.250    01-00-5e-7f-ff-fa    statisch

C:\Users\drews>

```

In anderen Netzwerk-Typen – also solchen, die nicht mit IP funktionieren – werden andere Techniken benutzt. Bei Novell-Netzen (IPX/SPX) wird die MAC-Adresse durch Zusatzinformationen erweitert und so eine Konnektivität zwischen Ethernet und der Prozessschicht (DoD-Modell) bzw. der ISO-OSI-Schicht 5 hergestellt.

Das Äquivalent zu ARP im neuen IPv6-Netzwerken ist das NDP (Neighbor Discovery Protocol)

**Aufgaben:**

1. Informieren Sie sich über die Optionen zum arp-Befehl innerhalb Ihres Betriebssystems!
- 2.
- 3.

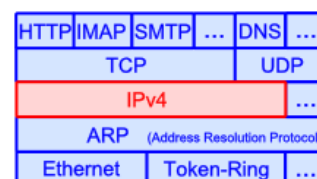
Da einige Einträge in der ARP-Tabelle von Programmen (statisch) eingetragen werden, besteht hier die Gefahr einer dauerhaften Schädigung / Beeinflussung des Netzwerk-Betriebes. Der Nutzer selbst kann den ARP-Cache nicht vollständig löschen. Durch **ARP-Spoofing** können vorsätzlich falsche MAC-Adressen im Netz verteilt werden. Dadurch sind dann im Netz Man-in-Middle-Angriffe möglich. Der Angreifer gibt sich als die Empfänger-Adresse aus, empfängt dann die Pakete und leitet diese nach dem Angreifen der interessierenden Daten an den eigentlich Empfänger weiter. Dessen MAC-Adresse hat sich der Angreifer gemerkt. Auch die Antworten des Empfängers an den ursprünglichen Sender werden auf die gleiche Tour abgegriffen.

**Internet Message Control Protokoll - ICMP**

Dieses Protokoll dient zum Austausch von Informationen und Fehlermeldungen. Die Version für das IPv6-Protokoll heißt ICMPv6.

Praktisch gehört das ICMP zu IPv4 oder eben zu IPv6 dazu. Es stellt quasi eine abgegrenzte Teilschicht dar, von der erwartet wird, dass sie ein Router oder ein Host auch separat versteht.

Es handelt sich um eine Verbindungs-lose Kommunikation (ähnlich UDP). Es findet also kein Handshake statt.



Fast immer handelt es sich bei ICMP-Paketen um Status-Informationen oder Fehler-Meldungen. Solche entstehen z.B. wenn Pakete vom Router nicht weitergeleitet wurden und somit zurückgewiesen wurden. Ein anderes Szenario sind abgelaufene Pakete. Jedes Paket erhält eine TTL-Nummer (Time-to-Live). Bei jedem Hop wird diese runtergezählt. Ist die TTL dann Null, wird das Paket verworfen (zerstört). Damit verhindert man, dass Pakete vielleicht unendlich lange im Internet herumgeistern, weil eine Ziel-Adresse auf einmal nicht mehr erreichbar ist.

ICMP hat keine eigene Pakete. Vielmehr werden angepasste IP-Pakete genutzt. Die wichtigen Bestandteile / Anpassungen sind im folgenden Datagramm rot gekennzeichnet.

Datagramm (eng.: Datagram)

Byte	+1		+2	+3	+4
0	IP-Version	IHL	Dienst-Art: <b>0000</b>	Paket-Länge	
4	Identifikation			Flag's	Fragment-Zähler
8	TTL		Protokoll: <b>0001</b>	Header-Kontrollsumme	
12	Quell-Adresse				
16	Quell-Adresse				
20	Ziel-Adresse				
24	Ziel-Adresse				
28	Optionen / Füll-Bit's				
32	<b>ICMP-Typ</b>	<b>ICMP-Code</b>		<b>ICMP-Prüfsumme</b>	
...	<b>ICMP-Daten (optional)</b>				

TTL .. Time to Live (Lebensdauer) wird bei jedem Hop um 1 verkleinert (→ dekrementiert), wenn 0 erreicht ist, dann wird Paket verworfen

Header-Checksum (Header-Kontrollsumme), wird bei jedem Hop neu berechnet (da sich ja z.B. die TTL jedesmal ändert!)

ICMP-Nachrichten sind immer spezifisch für einen Empfänger adressiert. Broadcast- oder Multicast-Nachrichten sind nicht möglich.

Typ	Typ-Name	Code	Code-Bedeutung
0	Echo (Antwort)	0	Echo (Antwort) (entspricht <b>pong</b> )
3	Ziel nicht erreichbar	0	Netzwerk nicht erreichbar
		1	Host nicht erreichbar
		2	Protokoll nicht erreichbar
		3	Port nicht erreichbar
		4	Fragmentierung nötig, aber <b>Don't Fragment-Flag</b> gesetzt
		5	Route nicht möglich
		13	Paket von Firewall geblockt
4	Entlasten der Quelle	0	Paket verworfen, da Warteschlange voll (Verkehr drosseln)
5	Umleitungs-Empfehlung zu anderem Gateway		
8	Echo (Anfrage)	0	Echo (Anfrage) (auch <b>ping</b> genannt)
9	Angebot eines Router's		
10	Router-Anwerbung		
11	Zeit-Limit überschritten	0	TTL (Time to Live) abgelaufen
		1	Zeitlimit während der Defragmentierung abgelaufen
12	Problem mit Paket-Parametern		
13	Zeitstempel		... für Synchronisation
14	Zeitstempel (Antwort)		
19	reserviert		... für Sicherheit
...			... für Robustheits-Experimente
29			
30	<b>Traceroute</b>		
40	Photuris-Protokoll		Sitzungsschlüssel Management Protokoll
42	erweitertes Echo		(Anfrage)
43	erweitertes Echo		Antwort

44	frei		
...			
252			
253	Experiment 1		
254	Experiment 2		

Für ICMPv6 gibt es abweichende Paket-Typen. Im Wesentlichen unterscheiden sich dabei nur die Typ-Nummern.

Typ	Typ-Name	Code	Code-Bedeutung
1	Ziel nicht erreichbar		Angabe der Komponente die nicht erreichbar ist
3	Zeit-Limit überschritten	0	TTL (Time to Live) abgelaufen
		1	Zeitlimit während der Defragmentierung abgelaufen
13	Zeitstempel		... für Synchronisation
128	Echo (Anfrage)	0	Echo (Anfrage) (auch <b>ping</b> genannt)
129	Echo (Antwort)	0	Echo (Antwort) (entspricht <b>pong</b> )
134	Angebot eines Router's		Router Advertisement
137	Redirect Message		Nachricht über die Umleitung eines Paket's

Das ICMP ist anfällig gegenüber DoS- und DDoS-Angriffen ((Distributed) Denial of Service). Desweiteren kann mittels ICMP-Tunnel ein unterschwelliger und meist unberechtigter Datenaustausch realisiert werden.

Traceroute-Programme senden ICMP-Nachrichten aus, die mit einer 1 als TTL und solange erhöht werden bis das Ziel erreicht ist. Für jedes Paket bekommt der Sender (Traceroute-Abfrager) von einem der Zwischenstationen (Hops) ein (Typ-11-)Paket zurück. Das bedeutet ja, dass die Lebenszeit abgelaufen ist. Dies wird der Paket-Quelle mitgeteilt. Aus den gesammelten Daten und den Antwortzeiten kann dann der Weg durch's Internet rekonstruiert werden.

Für die Überwachung eine Verbindung mit regelmäßigen Ping's können die folgenden Quell-Texte zurate gezogen werden.

In der Konsole kann schon mit:

```
ping -t Adresse
```

bzw.:

```
ping -t URL
```

eine laufende Kontrolle erfolgen.

Für VBScript eignet sich:

```
Function wmping(strComputer)
    Dim PingResults, Pingresult
    Set PingResults = GetObject("winmgmts://localhost/root/cimv2").
        ExecQuery("SELECT * FROM Win32_PingStatus WHERE
        Address = '" + strComputer + "'")
    For Each PingResult In PingResults
        If PingResult.StatusCode = 0 Then
            wmping = True
        Else
            wmping = False
        End If
    Next
End Function
```

Q: <https://www.msxfaq.de/tools/mswin/ping.htm>

---

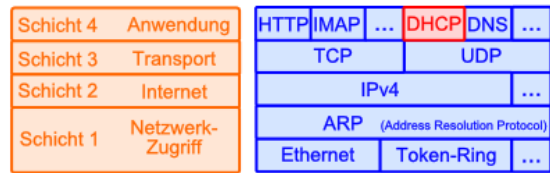
Auf der PowerShell kann das folgende Code-Schnipselchen benutzt werden. Dabei wird eine .Net-Klasse angerufen.

```
[string]$target= 127.0.0.1"  
$result = (new-object System.Net.NetworkInformation.Ping).Send($target)
```

Q: <https://www.msxfaq.de/tools/mswin/ping.htm>

## Dynamic Host Configuration Protokoll - DHCP

Ergänzung zum Bootstrap-Protokoll (BOOTP) zur Adresse-Zuordnung für Geräte ohne feste IP oder auch ohne Festplatten (lokale Betriebssysteme). Solche Diskless Workstation's bekommen alle Daten und Programme von Servern. Sie selbst sind praktisch nur noch Eingabe- und Ausgabe-Einheiten.



Heute wird DHCP aber auch in Netzen verwendet, um nicht jedem PC eine definierte IP zuzuweisen, was immer die Führung von Übersichten usw. notwendig macht. Für Rechner-Netze, bei denen die individuelle Host-Adresse im Netz egal ist, ist es eine der praktischen Lösungen.

Setzt eine Rechner voraus – meist ist das der zentrale Server – der eben diesen Dienst zur Verfügung stellt. Administrator muss im Allgemeinen nur den Adress-Bereich definieren und den Dienst starten. Optimalerweise ist der Bereich so gewählt, dass die maximale Anzahl gleichzeitig arbeitender Endgeräte sicher abgedeckt ist.

Zu kleine Bereiche bergen die Gefahr, dass irgendein Rechner keine IP mehr abbekommt und damit praktisch nicht im Netz arbeiten kann.

Zu große Bereiche ermöglichen es "böartigen" Clients (falsch eingesteckte oder fremd-Rechner) sich ins Netz zu integrieren, was ein Sicherheits-Risiko sein kann.

### **DHCP-Modi**

- manuelle Zuordnung**

feste Kombination von Client-MAC zu einer IP  
wird über eine Tabelle auf dem Server verwaltet
- automatische Zuordnung**

dauerhafte Zuordnung einer IP durch den Server zu einer MAC-Adresse  
wird in einer Tabelle auf dem Server dokumentiert
- dynamische Zuordnung**

zeitweilige Zuordnung einer IP durch den Server zu einer MAC-Adresse (Ausleihen einer IP)  
nach Ablauf einer Zeit (Lease-Time) muss spätestens eine neue IP beantragt werden

Byte	+1	+2	+3	+4
0	Operation	Netztyp	Adress-Länge	Relay-Anzahl
4	Verbindungs-ID			
8	Zeit seit Clientstart		Flag's	
12	Client-IP-Adresse			
16	eigene IP-Adresse			
20	Server-IP-Adresse			
24	Relay-Agent-IP-Adresse			
28	Client-MAC-Adresse			
32	Client-MAC-Adresse			
36	DHCP-Server-Name			
-	DHCP-Server-Name			
96	DHCP-Server-Name			
100	Datei-Name			
-	Datei-Name			
224	Datei-Name			
228	Optionen			
-	Optionen			
572	Optionen			

## DHCP-Operationen / - Nachrichten

- **DHCPDISCOVER** DHCP-Client sendet Anfrage als Broadcast an (alle) DHCP-Server im Netz
- **DHCPOFFER** DHCP-Server antwortet mit einem Angebot (auf eine Discover-Anfrage)
- **DHCPREQUEST** DHCP-Client fordert eine der angebotenen IP-Adressen von einem der DHCP-Server an ev. auch Verlängerungen der Release-Zeit, sowie weitere Daten (z.B. Gateway- und DNS-Server-Adressen)
- **DHCPACK** DHCP-Server bestätigt die Zuordnung der (angeforderten und angebotenen) IP-Adresse zu der MAC-Adresse des DHCP-Client's
- **DHCPNAK** DHCP-Server lehnt einen Request ab
- **DHCPDECLINE** DHCP-Client lehnt ein Ack ab, da die Adresse schon vergeben ist
- **DHCPRELEASE** DHCP-Client gibt seine Konfiguration frei (Server löscht die entsprechenden Einträge in seinen Tabellen)
- **DHCPINFORM** DHCP-Client fragt bei Server nach weiteren Konfigurations-Parametern (, wenn z.B. der Client eine feste IP besitzt)

Um z.B. IP-Adressen über Sub-Netze hinweg zu vergeben, können DHCP-Relay's benutzt werden. Damit lassen sich verfügbare Adressen über Subnetze hinweg noch effektiver und zentraler verwalten. DHCP-Relay's sind in diesen Fällen in Routern aktiv.

Es existiert zwar auch ein DHCPv6-Dienst, da aber keine IPv6 mehr verteilt werden, ist der Dienst für die Zuweisung von DNS-Servern und Gateway-Informationen zuständig.

DHCPv6 ermöglicht auch die Verteilung von weiteren Informationen zu Server-Diensten (z.B. NTP (Internet-Zeit) und SIP (Internet-Telefonie)). Die Kommunikation läuft über die UDP-Ports 546 für den Client und für den Server 547.

## DNS-Service

DNS steht für Domain-Name-Service.

Der DNS-Dienst (Dienst ist hier doppelgemoppelt) stellt quasi den Übersetzer zwischen den maschinen-orientierten IP-Adressen und den üblichen www-Adressen dar. Natürlich werden nicht nur die www-Adressen im Browser übersetzt sondern alle textuellen Domän-Adressen.

DNS steht praktisch auf Schicht 4 allen Anwendungen zur Verfügung.

Schon bei Einsatz von ping mit einem Servernamen kommt der DNS-Dienst zum Arbeiten.

Schicht 4	Anwendung	HTTP	IMAP	...	DHCP	DNS	...
Schicht 3	Transport	TCP		UDP			
Schicht 2	Internet	IPv4			...		
Schicht 1	Netzwerk-Zugriff	ARP (Address Resolution Protocol)					
		Ethernet	Token-Ring	...			

---

Dieses Mal wäre die Umsetztabelle aber einfach viel zu groß für unsere heimischen Rechner. Schließlich gibt es Milliarden von benannten Rechnern im Internet. Noch größer ist das Aktualisierungs-Problem. Die DNS-Tabelle kann sich mehrfach am Tag ändern, da müsste immer wieder eine neue Tabelle heruntergeladen werden. Diese würde auch zig Millionen von Adressen enthalten, die wir nie im Leben brauchen werden.

Die Macher von DNS haben deshalb den Dienst etwas anders eingerichtet. Bei der Netzwerk-Einrichtung müssen wir einen Rechner (dessen IP-Adresse) eingeben, der die Übersetzung machen soll, der also den DNS-Dienst bereitstellt. Bei DHCP (→ [Dynamic Host Configuration Protokoll - DHCP](#)) kann diese Zuweisung automatisiert erfolgen. Viele Internet-Provider geben bestimmte DNS-Server vor.

Ein universeller DNS-Server ist der von google: 8.8.8.8. Zum Ausprobieren ist der auch ok. Später sollte man sich einen heimischen Server wählen oder einer der anonymen.

Soll nun ein Domain-Name (z.B. google.de) in eine IP-Adresse aufgelöst werden, dann wird der eingerichtete DNS-Server befragt. Der schaut in seiner Tabelle nach und findet entweder einen passenden Eintrag, den er dann zurück sendet, oder er gibt die Anfrage an einen übergeordneten Server weiter. Genau der ist in seiner Netzwerk-Konfiguration als DNS-Server eingetragen.

Auf diese Art und Weise können die regionalen DNS-Server ihre Tabellen separat aktualisieren. Neue Adressen werden regional eingepflegt. Bekommt ein untergeordneter (unwissender) DNS-Server von einem übergeordneten Server eine passende Antwort zurückgeliefert, dann baut er diesen Eintrag in seine DNS-Tabelle ein.

Eine einfache Umwandlung und gleichzeitige Kontakt-Prüfung kann mit dem ping-Befehl erfolgen. Domain-Namen werden hier auch in IP-Adressen umgesetzt.

Um den Namen zu einer Adresse herauszubekommen kann man die Browser nutzen. Sie lassen auch die Eingabe einer IP-Adresse zu und wandeln diese dann aber in den Domainnamen mit der Homepage des Servers um.

Weitere – z.T. – genauere Informationen zum DNS findet der Leser auch noch im Abschnitt (→ [DNS – Domain Name Service](#) )

### **Aufgaben:**

#### ***1. Ermitteln Sie die IP-Adressen für die folgenden Website's!***

- |                     |                     |                     |
|---------------------|---------------------|---------------------|
| a) zdf.de           | b) www.bsi.bund.de  | c) www.atomzeit.eu  |
| d) de.wikipedia.org | e) en.wikipedia.org | f) dk.wikipedia.org |
| g) open.hpi.de      | h)                  | i)                  |

#### ***2. Wer steckt hinter den folgenden IP-Adressen? Gibt es sie überhaupt?***

- |                   |                   |            |
|-------------------|-------------------|------------|
| a) 52.178.155.90  | b) 212.227.247.48 | c) 8.8.8.8 |
| d) 91.198.174.192 | e) 192.168.0.1    | f) 2.2.2.2 |
| g) 127.0.0.1      | h) 153.384.245.16 | i) 1.1.1.1 |

#### ***3. Vergleichen Sie die (Telefon-)Auskunft mit dem DNS!***

#### ***für die gehobene Anspruchsebene:***

#### ***4. Prüfen Sie, ob Ihr Netzwerk auch mit IPv6 umgehen kann! Wenn JA, dann ermitteln Sie die IPv6-Adresse von google.de!***



Die Funktionalität des DNS-Dienstes kann man mit dem Konsolen-Befehl **nslookup** überprüfen. Das nslookup steht hierbei für "name service look up" – also "beim Namensserver nachschlagen".

Die einfache Eingabe einer IP-Adresse oder eines Domain-Namen's liefert den anderen zurück. Da die Server mehrere Netz-Anschlüsse haben (können) sind häufig auch mehrere DNS-Einträge verfügbar. IPv4 und IPv6 sind dabei völlig unabhängig.

zuerst wird auch noch der benutzte / angesprochene DNS-Server aufgelistet. Bei mir war das eine Fritz!-Box, die die Adresse 192.168.100.2 hatte.

Natürlich wusste meine Fritz!-Box nicht wirklich, welche Adressen mit welchen Namen assoziiert sind. Sie hat die Anfrage einfach an einen anderen Server weitergeschickt, der in der Konfiguration meiner Fritz!-Box hinterlegt war.

Die hinterlegten Namen können wiederum auch mal von den virtuellen Namen (hier: "google.de") abweichen und den physischen Rechner charakterisieren. Die beiden unteren Konsolen-Mitschnitte zeigen zumindestens den gleichen Zielrechner an, unabhängig davon, ob dieser über seine IPv4- oder die IPv6-Adresse angesprochen wird.

Da wir nun wissen, dass unser DNS funktioniert, können wir auch externe Server direkt abfragen. Der nslookup-Befehl lässt einige Optionen zu:

```
Eingabeaufforderung
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\lspdr>nslookup google.de
Server: fritz.box
Address: 192.168.100.2

Nicht autorisierende Antwort:
Name: google.de
Addresses: 2a00:1450:4014:800::2003
          216.58.201.67

C:\Users\lspdr>
```

```
Eingabeaufforderung

Name: google.de
Addresses: 2a00:1450:4014:800::2003
          216.58.201.67

C:\Users\lspdr>nslookup 2a00:1450:4014:800::2003
Server: fritz.box
Address: 192.168.100.2

Name: prg03s01-in-x03.1e100.net
Address: 2a00:1450:4014:800::2003

C:\Users\lspdr>
```

```
Eingabeaufforderung

Name: prg03s01-in-x03.1e100.net
Address: 2a00:1450:4014:800::2003

C:\Users\lspdr>nslookup 216.58.201.67
Server: fritz.box
Address: 192.168.100.2

Name: prg03s01-in-f3.1e100.net
Address: 216.58.201.67

C:\Users\lspdr>
```

```
Eingabeaufforderung

C:\Users\lspdr>nslookup /?
Syntax:
nslookup [-opt ...] # interaktiver Modus, Standardserver wird
                    verwendet.
nslookup [-opt ...] - server # interaktiver Modus, der Server "server"
                             wird verwendet.
nslookup [-opt ...] host # Der Host "host" wird gesucht, Standard-
                           server wird verwendet
nslookup [-opt ...] host server # Der Host "host" wird gesucht, der Server
                                "server" verwendet.

C:\Users\lspdr>
```

Testen wir das mal mit dem Test-DNS-Server 8.8.8.8 von google.

Gleich in der ersten Zeile sehen wir, dass wirklich der dns-Dienst von google geantwortet hat und zur Domäne zdf.de auch wirklich eine IP-Adresse verzeichnet hat.

```
Eingabeaufforderung

C:\Users\lspdr>nslookup zdf.de 8.8.8.8
Server: dns.google
Address: 8.8.8.8

Nicht autorisierende Antwort:
Name: zdf.de
Address: 91.197.29.78

C:\Users\lspdr>
```



---

## http- und https-Protokoll

Hypertext-Transport-Protokoll

Zustands-loses Protokoll →

kein Informations-Austausch zwischen zwei http-Anfragen

der Server hat keine Informationen darüber, ob er schon einmal angesprochen wurde

ist eine Identifikation notwendig, dann muss sie jedes Mal neu erfolgen

um dies bei bestimmten Webseiten (mit vielen http-Anfragen) zu unterdrücken, werden Cookies benutzt

Cookies sind kleine Text-Dateien, die Nutzer- und Kommunikations-Informationen enthalten

https ist das Sicherheits-Protokoll von HTTP

Client kann beim Server einen Komprimierungs-Modus für die Daten beantragen

für Texte sehr effektiv, bei vielen Bildern und Video's wegen der Eigen-Komprimierung wenig sinnvoll

Webseiten-Anfrage über die Seiten-URL

zusätzlich Query-Strings möglich; damit können zusätzliche Parameter an den Server gesendet werden

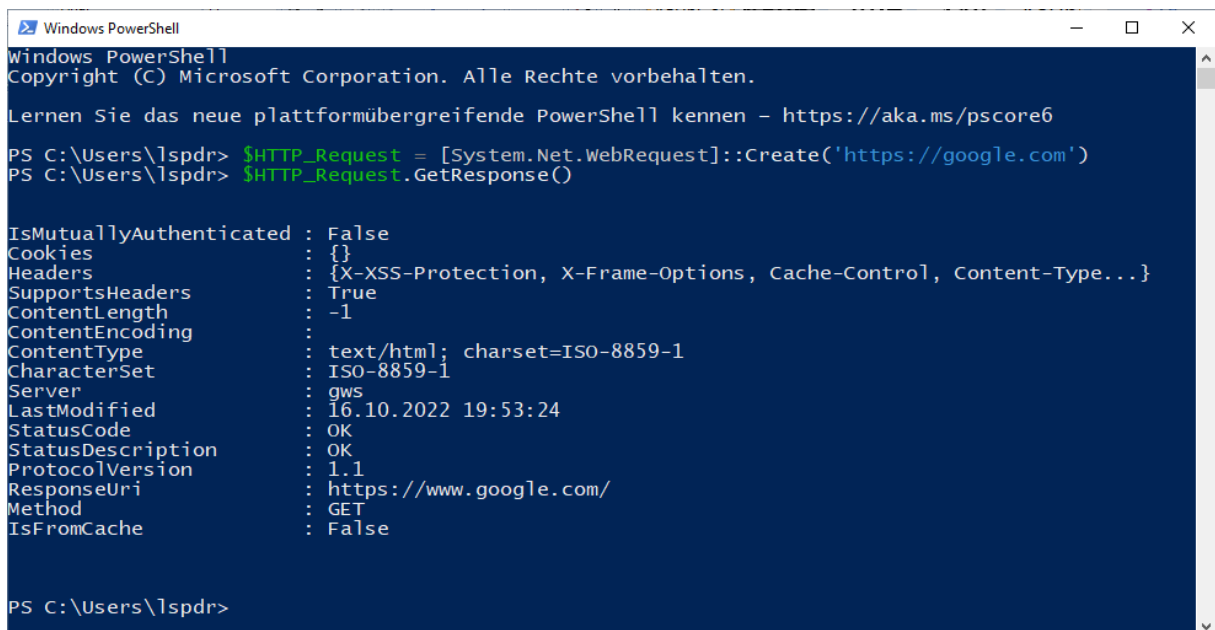
<https://google.de/search?q=informatik>

<https://amazon.com/s?k=informatik>

[https://google.com/s?k=informatik&&lr=lang\\_de](https://google.com/s?k=informatik&&lr=lang_de)

powershell öffnen (z.B. über [ Windows ] + [ R ])

dann die beiden Befehle eingeben



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Lernen Sie das neue plattformübergreifende PowerShell kennen - https://aka.ms/pscore6

PS C:\Users\lspdr> $HTTP_Request = [System.Net.WebRequest]::Create('https://google.com')
PS C:\Users\lspdr> $HTTP_Request.GetResponse()

IsMutuallyAuthenticated : False
Cookies                  : {}
Headers                  : {X-XSS-Protection, X-Frame-Options, Cache-Control, Content-Type...}
SupportsHeaders          : True
ContentLength            : -1
ContentEncoding          :
ContentType              : text/html; charset=ISO-8859-1
CharacterSet             : ISO-8859-1
Server                   : gws
LastModified             : 16.10.2022 19:53:24
StatusCode               : OK
StatusDescription        : OK
ProtocolVersion          : 1.1
ResponseUri              : https://www.google.com/
Method                   : GET
IsFromCache              : False

PS C:\Users\lspdr>
```

### **Response-Informationen**

- **IsMutuallyAuthenticated** haben sich Server und Client gegenseitig identifiziert

• <b>Cookies</b>	Liste der benutzten Cookies
• <b>Headers</b>	gibt darüber Auskunft, welche Funktionen der Server zur Verfügung stellt
• <b>SupportHeaders</b>	
• <b>ContentLength</b>	Länge / Größe des Inhalts-Teil's
• <b>ContentEncoding</b>	
• <b>ContentType</b>	Typ des Inhalts-Teil's
• <b>CharacterSet</b>	benutzte Zeichen-Tabelle
• <b>Server</b>	Name des Servers
• <b>LastModified</b>	gibt die letzte Aktualisierung der Webseite zurück
• <b>StatusCode</b>	Informationen zum Status des Webseiten-Aufruf's
• <b>StatusDescription</b>	Umschreibung der Status-Information
• <b>ProtokollVersion</b>	benutzte Version des Protokoll's
• <b>ResponseUri</b>	benutzte / angefragte URL
• <b>Method</b>	benutzte Methode des Protokoll's
• <b>IsFromCache</b>	

```
PS C:\Users\lspdr> [int]$HTTP_Request.GetResponse().StatusCode
200
PS C:\Users\lspdr>
```

das Beenden des Aufruf's erfolgt mit:

```
$HTTP_Response.Close()
```

damit wird die – ansonsten noch offene – Verbindung geschlossen

Staus-Code	Bedeutung / Text	Bemerkungen
<b>1xx</b>	Informationen	
<b>2xx</b>	erfolgreiche Operationen	
<b>200</b>	OK	
<b>204</b>	kein Inhalt	
<b>3xx</b>	Umleitungen	
<b>308</b>	Umleitung	
<b>4xx</b>	Clientfehler	
<b>400</b>	fehlerhafte Anfrage	
<b>403</b>	verboten	
<b>404</b>	nicht gefunden	
<b>5xx</b>	Serverfehler	

---

## **http-Anfrage-Methoden**

• <b>GET</b>	"bekommen" klassische Abfrage einer Webseite wird am häufigsten benutzt
• <b>HEAD</b>	"Kopf" liefert eine Kurzfassung der Daten, ohne die Daten selbst zu laden oft benutzt, um zu prüfen, ob Daten im Cache aktuell sind oder (vom Server) nachgeladen werden müssen
• <b>POST</b>	"Post" sendet Daten an den Server; z.B. mehrere Formular-Eingaben gebündelt
• <b>PUT</b>	"anlegen" zum Senden größerer Daten-Pakete an den Server; z.B. bei einem Bilder-Upload
• <b>DELETE</b>	"löschen" löscht Daten auf dem Server
• <b>OPTIONS</b>	"Möglichkeiten" zeigt die vom Server unterstützten http-Methoden an
• <b>CONNECT</b>	"verbinden"  wird bei Proxy-Servern genutzt
• <b>TRACE</b>	"verfolgen" sendet eine Anfrage so zurück, wie sie der Client abgeschickt hat häufig für Fehler-Suche benutzt
•	

über das **REST-Modell** lassen sich Daten – z.B. für verteilte Anwendungen – über das http-Protokoll zwischen Client und Server austauschen  
Representational State Transfer

# Codierung

Umsetzungs-Charakter überwiegt  
als einfach Umsetzung verstanden  
ev. noch Redundanzen oder Absicherungen () ergänzt

<b>Definition(en): Codierung</b>
Codierung ist die Anwendung eines Codes (Kodes) auf ein einzelnes Zeichen oder eine Zeichenfolge zur Erzeugung einer neuen Signalfolge.
Codierung ist die Umsetzung von Daten (Signale, Zeichen, ...) in eine andere – für die speziellen Anwendungen – geeignete Signale, Zeichen, ...
Im Bereich der Software-Erstellung versteht man unter Codierung auch das Umsetzen eines Algorithmus in ein Computerprogramm (Programm-Erstellung / Programm-Entwicklung).

wenn Geheimhaltungs-Charakter überwiegt, dann wird eher von Chiffrierung / Verschlüsselung gesprochen

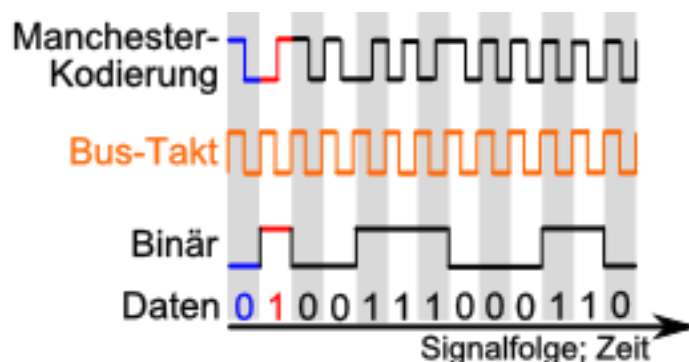
## Codierung von 0 und 1 auf Schicht 1

### Manchester-Kodierung

DEMBOWSKI, S. 81

klassisches Daten-Kodierungs-Verfahren in älteren Netzwerken (z.B. 10 Mbit/s-Ethernet)

Basis ist Rechteck-Takt-Signal für die Übertragung eines Bits wird genau eine Periode benutzt  
erste Hälfte der Periode stellt das invertierte Bit dar, die zweite das ursprüngliche Bit  
dadurch ist für jedes Bit ein Pegelwechsel auf der Leitung realisiert  
Pegel-loses und Takt-freies Medium bedeutet freies Medium (→ CSMA/CD-Verfahren)



Effizienz liegt bei 50%

moderne Verfahren (4B/5B-Kodierung) in schnelleren Netzen (100 Mbit/s Ethernet) bringen es auf 80%

## 4B/5B-Kodierung

eine 4-bit-Datenfolge wird durch einen 5-bit-Code verschlüsselt

Erhöhung der Übertragungssicherheit

überzählige Signalfolgen werden für Korrektur- und Steuer-Zwecke genutzt

weiterhin soll durch die relativ gleichmäßige Zahl von Nullen und Einsen die Entstehung von Gleichspannungs-Belastungen gering gehalten werden

<b>Daten</b> (Halb-Byte, Nibble, TX/RX- Bitfolge)	<b>Zeichen</b>	<b>Code</b>	<b>Bedeutung / Funktion</b>	<b>Bemerkung</b>
<b><i>Daten-Gruppe</i></b>				
0000	0	11110	Data 0	Zeichen 0
0001	1	01001	Data 1	Zeichen 1
0010	2	10100	Data 2	Zeichen 2
0011	3	10101	Data 3	Zeichen 3
0100	4	01010	Data 4	Zeichen 4
0101	5	01011	Data 5	Zeichen 5
0110	6	01110	Data 6	Zeichen 6
0111	7	01111	Data 7	Zeichen 7
1000	8	10010	Data 8	Zeichen 8
1001	9	10011	Data 9	Zeichen 9
1010	A	10110	Data A	Zeichen A
1011	B	10111	Data B	Zeichen B
1100	C	11010	Data C	Zeichen C
1101	D	11011	Data D	Zeichen D
1110	E	11100	Data E	Zeichen E
1111	F	11101	Data F	Zeichen F
<b><i>Kontroll-Gruppe</i></b>				
	I	11111	Idle: Stream Fill Code	Leerlauf, Füll-Code
0101	J	11000	Start of Stream Delimiter 1	Startsignal Teil 1
0101	K	10001	Start of Stream Delimiter 2	Startsignal Teil 2
	T	01101	End of Stream Delimiter 1	Endsignal Teil 1
	R	00111	End of Stream Delimiter 2	Endsignal Teil 2
<b><i>Fehler-Gruppe</i></b>				
	H	00100	Transmit Error	Übertragungs-Fehler
<b><i>Gruppe ungültiger Codes</i></b>				
	V	00000		ungültig
	V	00001		ungültig
	V	00010		ungültig
	V	00011		ungültig
	V	00100		ungültig
	V	00101		ungültig
	V	00110		ungültig
	V	01000		ungültig
	V	10000		ungültig
	V	11001		ungültig

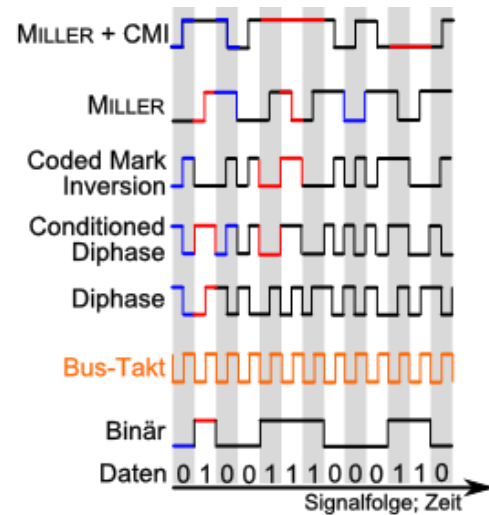
## 8B/10B-Kodierung

für schnellere Netze (ab 1'000 Mbit/s)

für ein Byte gibt es zwei Codes, die sich hinsichtlich der Anzahl von Nullen und Einsen bei  
Aufeinanderfolge immer ergänzen

Gleichspannungs-Belastung (dadurch) sehr gering

→ KALDEALI → S. 14 ff.



Q: geändert. nach /3/

---

## weitere Kodierungen

### MLT-3-Kodierung

arbeitet mit drei Pegeln (+V, 0V, -V)  
dadurch wird die Signal-Frequenz deutlich verringert

### PAM5- und Trellis-Kodierung

Multilevel-Kodierungen  
PAM5 verwendet fünf Pegelstufen

Trellis-Kodierung ergänzt 8 bits um ein Paritäts-Bit zur Erhöhung der Datensicherheit

PAM5- und Trellis-Kodierung kommt bei 1000BaseTX zum Einsatz

CRC usw.

→ KALDEALI → S. 11 ff.

Testen von Protokollen

→ KALDEALI → S. 27 ff.

## RC5-Code

Verwendung bei IR-Fernbedienungen von Haushalts-Elektronik (Fernseher, Radio, CD-Player, Video-Recorder, ...)

RC ... radio controlled

1980 von der Firma Philips entwickelt

klassischer RC5-Code besteht aus 14 bit

die ersten 3 Bits dienen der Start-Erkennung, Signalstärke-Abstimmung und der Erkennung von neuen Kommando im Vergleich zu Dauer-Komandos (Tastendruck-Erkennung)  
das Toggle-Bit (T) ändert seine Wert immer dann, wenn eine neue Taste gedrückt wird

es folgen 5 Adress-Bits, welche die Geräte-Klasse codieren (Fernseher, Video-Recorder, DVD-Player, ...)

die letzten 6 Bit sind Kommando-Bits  
sie enthalten den konkreten Steuer-Befehl – das Kommando

Code-Bits	1	1	T	A5	A4	A3	A2	A1	K6	K5	K4	K3	K2	K1
Codeteil	Steuerteil			Adressteil					Kommandoteil					

praktisch als  $2^5 = 32$  verschiedene Geräte-Typen möglich

### Geräte-Adressen

Adresse (dez)	Geräte-Typ	
0	Fernseher	
3	Videotext	
4	Laser-Video-Player	
5	Video-Recorder	
7	für Experimentierzwecke / Eigenbaus	
8	Sat-Receiver	
12	DVD-Player	
13	für Experimentierzwecke / Eigenbaus	
14	CD-Photo-Player	
16	Vorverstärker	
17	Radio-Tuner	
18	Cassetten-Recorder	
20	CD-Player	
21	Plattenspieler	
26	CD-Recorder	
29	Lichtsteuerung	



praktisch als  $2^6 = 64$  verschiedene Kommandos möglich

### allgemeine (gemeinsame) Kommandos

Kommando (dez)	Kommando	
0 ... 9	numerische Taste	
12	Standby	
13	Mute	
16	Volume +	
17	Volume -	
18	Brightness +	
19	Brightness -	
20	Color saturation +	
21	Color saturation -	
22	Bass +	
23	Bass -	
24	Treble +	
25	Treble -	
26	Balance right	
27	Balance left	
53	Play	
54	Stop	
63	System select	

erweiterter / moderner RC5-Code

zweites Steuer-Bit dient somit zur Unterscheidung des unteren (klassischen) Kommando-Sets von oberen (erweiterten / modernen) Kommando-Set

Code-Bits	1	¬K7	T	A5	A4	A3	A2	A1	K6	K5	K4	K3	K2	K1
Codeteil	Steuerteil			Adressteil					Kommandoteil					

somit insgesamt  $2^7 = 128$  Kommandos möglich, wobei nur der obere Bereich ( $\neg K7 = 0$ ) neu belegbar ist

### erweiterte Kommandos

Kommando (dez)	Kommando	
89	Ambilight (Hintergrundbeleuchtung)	

---

## Umsetzung einer (möglichen) Decodierung des RC5-Codes

### Prozeduren / Funktionen auf Decoder-Seite:

RC5ADRESSE

RC5BEFEHL

RC5STATUS (verwaltet verschiedene Register z.B. RC5TOGGLEALT (altes Toggle-Bit) und RC5TOGGLENEU (neues Toggle-Bit); RC5NEUETASTE (gesetzt wenn neue Taste gedrückt wurde und mehrmals das gleiche Telegramm empfangen wurde))

RC5TELGUELTIG (gibt an, ob der RC5-Code (Telegramm) gültig ist)

RC5ROUTINE (fragt alle 250 µs den IR-Empfänger auf ein neues Telegramm ab und speichert es)

RC5ACTIONx (Reaktion auf das Kommando x)

### Links / Quellen:

[http://www.stefan-buchgeher.info/elektronik/rc5/rc5\\_doku.pdf](http://www.stefan-buchgeher.info/elektronik/rc5/rc5_doku.pdf) (Decoder-Prozeduren, ...)

## Decabit-Impulsraster

Rundsteuersystem für Geräte über das Strom-Netz

Ende der 1960er Jahre entwickelt

Fa. Zellweger (heute Teil von ASCOM)

Impuls-Abstands-Verfahren (alternativ wäre in einem anderen Verfahren z.B. die Codierung über ein Impuls-Intervall-Verfahren möglich)

ein Steuer-Signal (ein Decabit-Signal) besteht aus einem Start-Impuls und 10 Steuer-Impulsen

insgesamt ist ein Kommando-Signal 6,6 s = 6'600 ms lang auf jeden Impulsteil verfallen dabei exakt 600 ms = 0,6 s

dabei werden immer 5 Daten-Impulse und 5 Pausen verwendet

dadurch eine 5-aus-10-Auswahl

ergibt insgesamt 126 Doppel-Kommando's

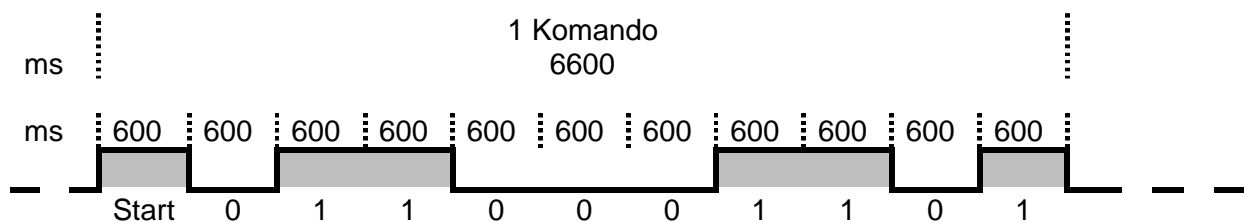
Code	Signal
0	0010111010
1	1001110010
2	1001101010
3	1001011010
4	0000111011
5	1100111000
6	1100110010
7	1100101010
8	1100011010
9	0001111010
10	1010111000
11	1010101010
12	1010011010
13	1000110110

Code	Signal
32	1010110100
33	1000111100
34	1010010110
35	1110011000
36	1110010100
37	1110001100
38	1100011100
39	0010111100
40	1100110100
41	0101010110
42	1100001110
43	1000010111
44	1100010101
45	1101010100

Code	Signal
64	1101001001
65	1101110000
66	1111001000
67	1001111000
68	0101111000
69	1101001010
70	0110001110
71	1000101110
72	0010101110
73	1000011110
74	0010011110
75	1110001010
76	1011000110
77	1001001110

Code	Signal
96	0110101010
97	0100110110
98	0001110110
99	0100101110
100	1000111010
101	0100111010
102	1010110010
103	1001101100
104	1101100100
105	1011001100
106	1010011100
107	0110011100
108	1100010110
109	1101000110

14	1000110011	46	1101010010	78	0011001110	110	1110100010
15	0011101100	47	1110000110	79	1010001011	111	1110100100
16	0001101110	48	1100100110	80	0111001010	112	1101011000
17	1000101101	49	1001010110	81	0101101010	113	1101101000
18	1001100101	50	1111000010	82	0111000110	114	1010001110
19	1001101001	51	1101100010	83	0101100110	115	1011001010
20	1011100100	52	1011100010	84	0100011110	116	0101001110
21	1001110100	53	0111100010	85	0111100100	117	0111010100
22	1100101100	54	1010100011	86	0110110100	118	1011010010
23	0101101100	55	1110110000	87	0011110100	119	0110111000
24	1001100110	56	1110101000	88	0011011100	120	1110010010
25	1011101000	57	1010100110	89	0011010110	121	1111100000
26	1101001100	58	0110100110	90	1011110000	122	0111110000
27	1010101100	59	1110100001	91	0011110000	123	0011111000
28	1001011100	60	1111010000	92	0011011010	124	0001111100
29	1001001101	61	0111011000	93	1001010011	125	0000111110
30	1011011000	62	0101011010	94	1011000011	126	1111111111
31	1011010100	63	1100011001	95	0101110010		



Steuer-Signale werden z.B. von einem Steuer-PC auf das Strom-Netz aufmoduliert  
verwendete Frequenzen liegen zwischen 167 und 2'000 Hz  
für Netze mit kleiner Ausdehnung werden Frequenzen über 250 Hz empfohlen, für größere  
Netz-Ausdehnungen die Frequenzen unter 250 Hz  
zusteuendes Gerät hängt an einem Rundsteuer-Empfänger, der den Strom für dieses Gerät  
freischaltet bzw. wieder abschaltet  
als Basis-Frequenz wird die normale Stromnetz-Frequenz von 50 Hz (Phasenlänge: 0,02 s =  
20 ms) benutzt  
Weiterentwicklungen der Fa. Zellweger arbeiten mit längeren und mehr Impulsen

praktische Umsetzungen  
SemagyrTOP, Versacom und Swistra (Fa. Swistec)

bei Impuls-Intervall-Verfahren

je nach Firmen-Umsetzung unterschiedliche Längen der Start- und Daten-Impulse  
auch Pausen zwischen Start und Daten sowie innerhalb der Daten sind unterschiedlich  
charakteristisch ist eine deutlich höhere Anzahl von Impulsen in jedem Protokoll

---

## **Unified Diagnostic Services (UDS)**

dt.: Vereinheitlichte Diagnose-Dienste

vorrangig im Automobil-Bau verbreitet, aber in abgewandelter Form auch in Flugzeugen, Schiffen, U-Booten usw.

Ziel ist der einheitliche Zugriff auf Diagnose-Informationen aus den Steuergeräten der Fahrzeuge unabhängig vom Hersteller (des Steuergerätes und des Fahrzeuges)

UDS-Nachricht besteht immer aus einem SID-Feld (Service-ID), einem Parameter-Feld und einem Daten-Feld

Kommunikation ist Verbindungs-orientiert (Anfrage-Antwort-Kommunikation)

---

## **Chiffrierung**

→ KALDEALI → S. 26

Codierung mit Betonung auf Geheimhaltung bzw. Unlesbarmachung

<b>Definition(en): Chiffrierung</b>
Chiffrierung / Verschlüsselung ist die Umwandlung eines Zeichen oder einer Zeichenfolge ("Klartext") mittels eines Schlüssels in ein Chifftrat ("Geheimtext").
Chiffrierung ist eine Codierung, bei der das Zeichen oder die Zeichenfolge bis zur Dechiffrierung unkenntlich gemacht wird.

### **Links:**

<http://users.telenet.be/d.rijmenants/en/enigmasim.htm> (Enigma-Simulator; engl.)

---

## 4. praktische Netzwerke und ihre Protokolle

### 4.1. das Ethernet

HTTP	IMAP	SMTP	...	DNS	...
TCP				UDP	
IPv4					...
ARP (Address Resolution Protocol)					
Ethernet		Token-Ring		...	

ganz ursprünglich funktionierten alle Computer-Verbindungen über Funk  
daher auch die Bezeichnung ether für Äther

dann aber für lokale Vernetzung LAN spezifiziert

im OSI-Modell Layer 1 (physische Schicht) und Layer 2 (Data-Link-Schicht)

#### **Definition(en): Ethernet**

Unter Ethernet versteht man die Technologie, Hardware und Software für die (vorrangig) lokale Vernetzung von Computern und der zugehöriger Perepherie.

#### **Priorisierung in Heim-Netzwerken mit FRITZ!-Routern**

"Internet" → "Filter" → "Priorisierung"

auch für VPN's sinnvoll

---

## Standard-Ethernet

### 10Base5

Thick-Ethernet

10 MBit/s

dickes, gelbes Koaxial-Kabel → yellow cable

Bus-Topologie

Segment max. 500m weit; max. 100 Stationen zugelassen

zu jeder Station gehört ein externer Transceiver, über AUI-Kabel mit der Netzwerkkarte der Station verbunden

Stationen können entfernt werden, Netzwerk funktioniert weiterhin ordnungsgemäß

relativ teuer (Kabel, Transceiver), Kabel-Beschädigungen bewirken Ausfall des gesamten Netzwerks

### 10Base2

Cheapernet, Thin wire, thin cable

10 MBit/s

Segment max. 185m weit; max. 30 Stationen zugelassen

Bus-Topologie

Anschluß über T-Verbinder

Stationen können nicht entfernt werden ohne dass das Netz zusammenbricht; Kabel-Beschädigungen bewirken Ausfall des gesamten Netzwerks

preiswert (Netzwerkkarte enthält Transceiver, billigeres Kabel); Einkabel-Anbindung ganzer Netze möglich (wenig Verkabelungsaufwand in Bauwerken)

### 10BaseT

Stern-Topologie, Punkt-zu-Punkt

Twisted Pair-Verkabelung (**Cat. 3**, 4 od. 5) (RJ45-Stecker)

Segment max. 100m weit, Anzahl der nicht direkt begrenzt

Stationen frei austauschbar, Kabel-Beschädigungen wirken sich nur auf eine Verbindung aus bei Einsatz eines Hub nur immer eine aktive Verbindung möglich, mit Einsatz von Switches lassen sich viele parallele Verbindungen aufrechterhalten

### 10BaseF

wie 10BaseT nur statt der Kupfer-Kabel werden Glasfaser-Leitungen (Lichtwellenleiter, LWL) verwendet

Segment kann nun 2km weit sein

bestehende LWL-Vernetzung kann auch für schnellere Verbindungen genutzt werden (also sehr Zukunftssicher)

Verarbeitung der LWL aufwendiger und damit teurer

---

## Fast-Ethernet

100Base

## Gigabit-Ethernet



---

## zukünftige Ethernets

nur im professionellen Umfeld, Hochleistungs-Cluster, Rechenzentren

2,5GBaseT

5GBaseT

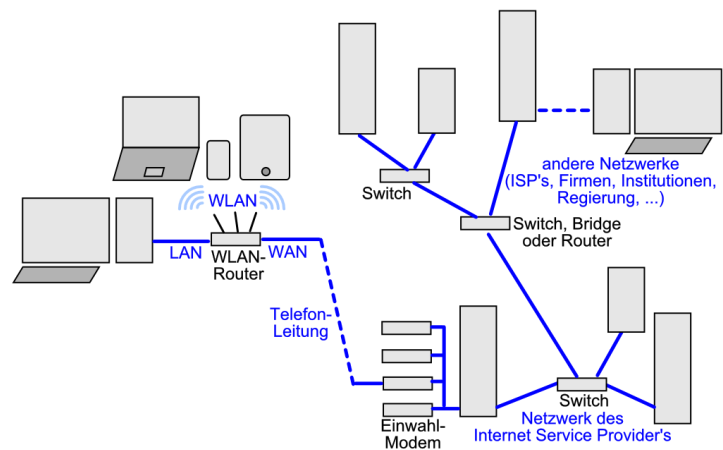
10GBaseE

weiter in Arbeit 25G-, 40G-, 50G-, 100G-, 400G- und 1TBase-Ethernet

### Aufbau eines Ethernet-Paketes (mit maximalen IPv4- / TCP-Daten)

Schicht 4: TCP-Segment								TCP-Header	Nutzdaten 1460 Byte		
Schicht 3: IP-Paket							IP-Header	Nutzdaten 1480 Byte			
Schicht 2: Ethernet-Frame			Empfänger-MAC	Absender-MAC	802.1Q-Tag (opt.)	Ethernet-Typ (0x0800)	Nutzdaten 1500 Byte		Frame Check Sequence		
Schicht 1: Ethernet-Paket+IPG	Präambel	Start of Frame	Nutzdaten 1518 / 1522 Byte								Interpacket Gap
Anzahl Byte's (Oktette)	7	1	6	6	(4)	2	20	20	6 - 1460	4	12

## 4.1.x. Ethernet-Geräte



## 2.5.x.y. Repeater, Hub's, Switches, Router, Bridges, Gateway's

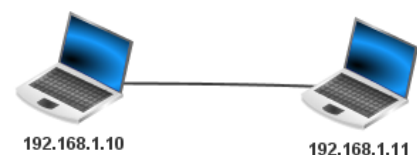
gemeint hier i.A. Geräte  
entweder als Einzel-Geräte, die vielfach für den professionellen Einsatz gedacht sind  
oder Kombi-Geräte; eher für den heimischen Bedarf  
meist Vielzahl von Netzwerk-Anschlüssen ((Netzwerk-)Port's) mit eigenen MAC-Adressen

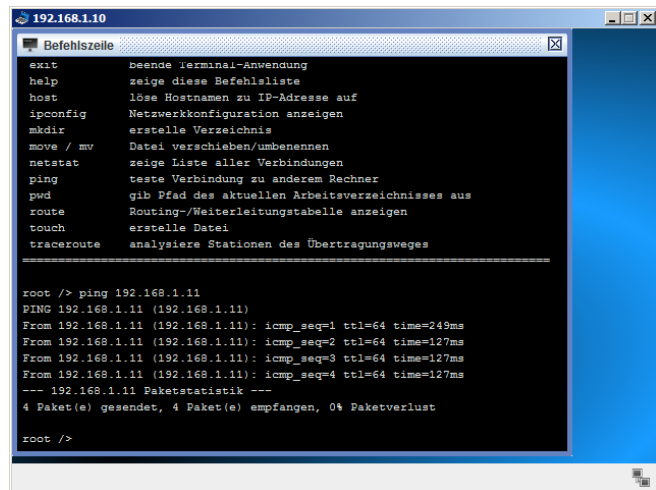
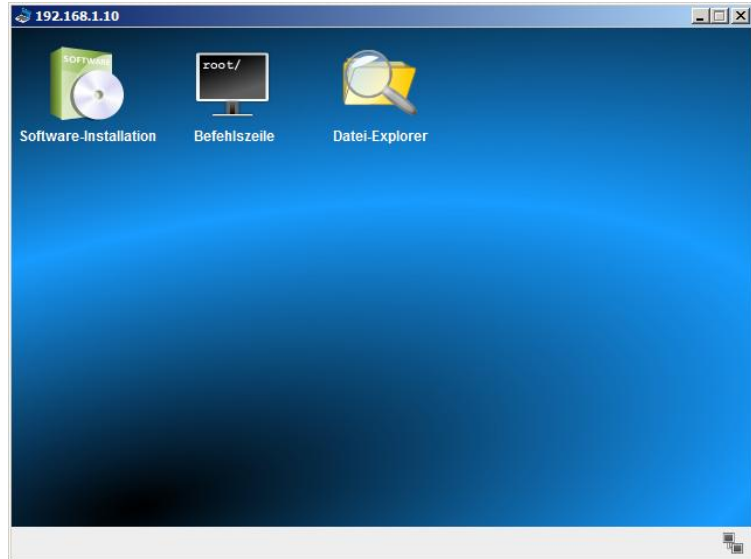
nur wenige Geräte benötigen eigene IP-Adressen  
z.B. Router, Gateway's

bessere Geräte mit einer gewissen Eigen-Intelligenz verfügen über Anwendungs-spezifische integrierte Schaltung (ASIC's) (Hardware-Intelligenz )  
optimierte Schaltkreise für den Netzwerk-/Daten-Verkehr

einige Geräte verfügen über (eigene) Betriebssysteme  
dazu gehören Router und Gateway's

viele Leistungen heute auch als Software möglich, bis hin zur (vollständigen) Virtualisierung von Netzen



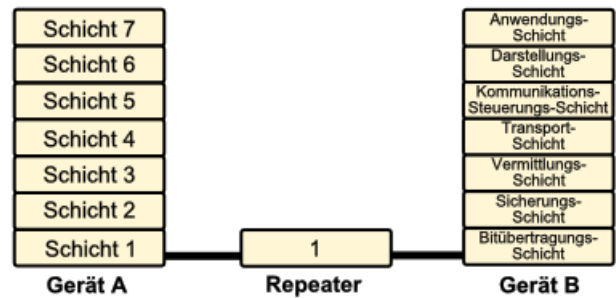


Nr.	Zeit	Quelle	Ziel	Protokoll	Schicht	Bemerkungen
1	19:49:56.364	192.168.1.10	192.168.1.11	ARP	Vermittlung	Suche nach MAC für 192.168.1.11, 192.168.1.10: 69:3F:DD:C6...
2	19:49:56.488	192.168.1.11	192.168.1.10	ARP	Vermittlung	192.168.1.11: AC:E7:C9:BA:9B:8F
3	19:49:56.488	192.168.1.10	192.168.1.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 1
4	19:49:56.613	192.168.1.11	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 1
5	19:49:57.567	192.168.1.10	192.168.1.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 2
6	19:49:57.694	192.168.1.11	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 2
7	19:49:58.770	192.168.1.10	192.168.1.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 3
8	19:49:58.897	192.168.1.11	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 3
9	19:49:59.973	192.168.1.10	192.168.1.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 4
10	19:50:00.100	192.168.1.11	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 4

## 2.5.x.y.z. Repeater

Wiederholer – praktisch Verstärker bzw. Regenerator des Signals

arbeitet – wie Netzkabel bzw. Medien – auf OSI-Schicht 1 (Bit-Übertragung)



### **Repeater-Varianten**

- **Transceiver**                      Kombinationsgerät aus Transmitter (Sender) und Receiver (Empfänger)
- **Sternkoppler**                    einfacher Zusammenschluss mehrerer Leitungen / Kabel / Medien
- **Hub**                                Gerät zur sternförmigen Verbindung von Netzwerk-Geräten bzw. Netzwerken (→ Hub)
- **Medien-Konverter**                einfache Version ohne Bridge-Funktion (→ )

### **Definition(en): Repeater**

Ein Repeater ist ein Netzwerkgerät zur Weiterleitung / Verstärkung / Regeneration eines Signals.

### **Aufgaben:**

1.

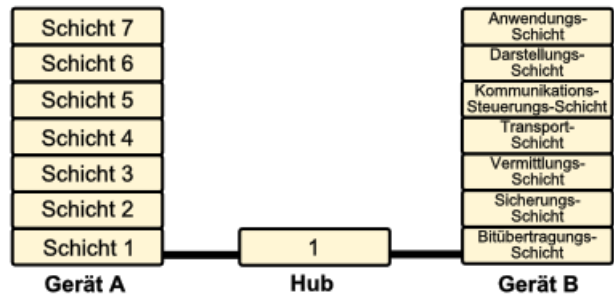
**für die gehobene Anspruchsebene:**

***x. Informieren Sie sich über die 5-4-3-Regel (Repeater-Regel)! Was besagt sie und welcher technische Zweck / Grund steckt dahinter?***

## 2.5.x.y.z. Hub

auch Multi-Port-Repeater

ist ein Repeater mit mehreren Netzwerk-Anschlüssen auf der einen Seite arbeitet auf OSI-Schicht 1 (Bit-Übertragung)



Signale werden auf alle Port weitergeleitet, dadurch praktisch immer nur eine Verbindung über den Hub möglich  
stellt Engpass dar (besonders zum Server hin)

- 
- 
- 

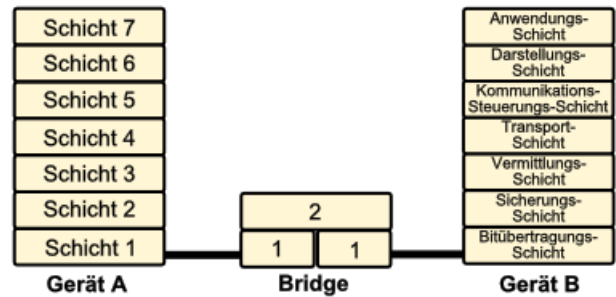
meist nur 10MB-Hubs verbaut  
theoretisch nur bis 1GB-LAN verfügbar  
praktisch nur selten 100MB-Hubs vorhanden

### **Definition(en): Hub**

Ein Hub ist ein Netzwerk-Gerät, mit dem andere Netzwerk-Geräte auf physikalischer Ebene sternförmig verbunden werden.

## 2.5.x.y.z. Bridge

arbeitet auf OSI-Schicht 2 (Sicherungs-Schicht)



- 
- 
- 

eine Bridge ist transparent, d.h. sie ist im Netz unsichtbar, hat also z.B. keine IP-Adresse

### **Definition(en): Bridge**

Eine Bridge ist ein Netzwerk-Knoten, der zwei gleichartige Netzwerk-Segmente miteinander verbindet / koppelt (im Sinne einer Verlängerung / Ausweitung).

## 2.5.x.y.z. Switch

Netzwerkweiche, Verteiler  
Umschalter, Schalter

Multi-Port-Bridge



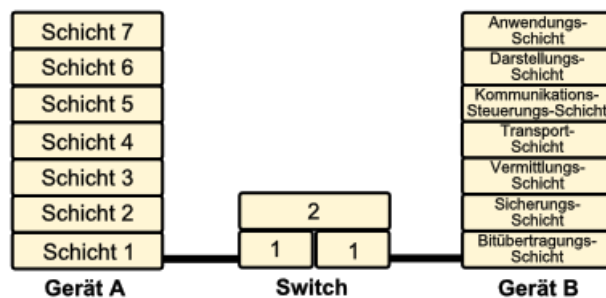
Verbindung von Netzwerk-Segmenten

Cisco-Symbol  
für ein Switch  
Q: de.wikipedia.org  
(Deadlyhappen)

arbeitet auf OSI-Schicht 2 (Sicherungs-  
Schicht) → Layer-2-Switch

gemanagte Switche arbeiten auch auf  
Schicht 3 oder noch darüber (Layer-3-  
Switch)

Switche dieser Kategorie übernehmen  
zusätzliche Funktionen (IP-Filterung,  
Quality of Service, ...)



Switche merken sich die Verbindungen / Ports und stellen dann parallel mehrere Verbindungen gleichzeitig her; Gerät lernt die Hardware-Netzwerkadressen (MAC) der Einzel-Geräte  
Pakete werden nur weitergeleitet, wenn Empfänger verfügbar ist



- 
- 
- 

#### Vorteile

mehrere Verbindungen gleichzeitig möglich

schnellere Vermittlung der Einzel-Verbindungen

Voll-Duplex-Betrieb möglich

Ports können zusammengefasst / gekoppelt werden und damit ein höherer Datendurchsatz erreicht werden

bei L-3-Switches ist die Austeilung des LAN in virtuelle LANs (VLANs) möglich, die voneinander abgeschottet werden können

#### Nachteile

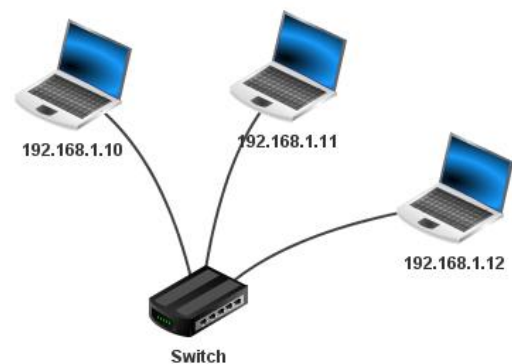
es ist schwieriger einen Fehler zu finden, da Pakete nur noch auf bestimmten Leitungen gehandelt werden

größere Latenzzeiten (weil das Gerät aktiv am Paket-Transport und –Vermittlung mitwirkt)

von sich aus keine Redundanz (Probleme bei Fehl-Funktionen oder Ausfällen)

### Definition(en): Switch

Ein Switch ist ein Netzwerk-Gerät zur sternförmigen Verbindung von eigenständigen Netzwerk-Segmenten.



```

192.168.1.10
Befehlszeile
exit      beende Terminal-Anwendung
help      zeige diese Befehlsliste
host      löse Hostnamen zu IP-Adresse auf
ipconfig  Netzwerkkonfiguration anzeigen
mkdir     erstelle Verzeichnis
move / mv Datei verschieben/umbenennen
netstat   zeige Liste aller Verbindungen
ping      teste Verbindung zu anderem Rechner
pwd       gib Pfad des aktuellen Arbeitsverzeichnisses aus
route     Routing-/Weiterleitungstabelle anzeigen
touch     erstelle Datei
tracert   analysiere Stationen des Übertragungsweges

=====

root /> ping 192.168.1.12
PING 192.168.1.12 (192.168.1.12):
From 192.168.1.12 (192.168.1.12): icmp_seq=1 ttl=64 time=4119ms
From 192.168.1.12 (192.168.1.12): icmp_seq=2 ttl=64 time=2059ms
From 192.168.1.12 (192.168.1.12): icmp_seq=3 ttl=64 time=2059ms
From 192.168.1.12 (192.168.1.12): icmp_seq=4 ttl=64 time=2059ms
--- 192.168.1.12 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root /> |

```

Nr.	Zeit	Quelle	Ziel	Protokoll	Schicht	Bemerkungen
1	19:56:45.316	192.168.1.10	192.168.1.11	ARP	Vermittlung	Suche nach MAC für 192.168.1.11, 192.168.1.10: 69:3F:DD:C...
2	19:56:45.440	192.168.1.11	192.168.1.10	ARP	Vermittlung	192.168.1.11: AC:E7:C9:BA:9B:8F
3	19:56:45.440	192.168.1.10	192.168.1.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 1
4	19:56:45.565	192.168.1.11	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 1
5	19:56:46.519	192.168.1.10	192.168.1.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 2
6	19:56:46.646	192.168.1.11	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 2
7	19:56:47.722	192.168.1.10	192.168.1.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 3
8	19:56:47.849	192.168.1.11	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 3
9	19:56:48.925	192.168.1.10	192.168.1.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 4
10	19:56:49.052	192.168.1.11	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 4
11	20:35:16.909	192.168.1.10	192.168.1.12	ARP	Vermittlung	Suche nach MAC für 192.168.1.12, 192.168.1.10: 69:3F:DD:C...
12	20:35:18.969	192.168.1.12	192.168.1.10	ARP	Vermittlung	192.168.1.12: DB:9B:D8:3D:4C:98
13	20:35:18.969	192.168.1.10	192.168.1.12	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 1
14	20:35:21.028	192.168.1.12	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 1
15	20:35:21.231	192.168.1.10	192.168.1.12	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 2
16	20:35:23.290	192.168.1.12	192.168.1.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 2

Unter Switching versteht man die geordnete Weiterleitung von Daten(-Paketen) auf vorbestimmten / bekannten Wegen. Ist dieser unbekannt, dann wird er vor der Weiterleitung erst ermittelt.

## 2.5.x.y.z. Router

spricht: rauter; neudeutsch: ruhter

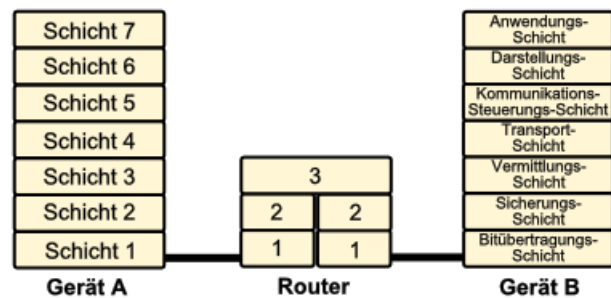
Addressierung über das eigene Netz hinaus  
Anpassung PDU-Struktur (Protocol Data Unit)  
Anpassung der Protokoll-Parameter  
Anpassung der Fehlerbehandlungs-Mechanismen  
Abbildung von Diensten  
Abbildung des lokalen Wegwahl-Mechanismus auf globale Wegewahl



Cisco-Symbol  
für einen Router  
Q: de.wikipedia.org  
(George Shuklin)

arbeitet auf OSI-Schicht 3 (Vermittlungs-  
Schicht)

verbinden verschiedene Netze



- 
- 
- 

### **Definition(en): Router**

Router sind Netzwerkgeräte, die Daten(-Pakete) zwischen verschiedenen Netzen austauschen können.

Ein Router ist eine Node, der Daten(-Pakete) weiterleitet, die nicht für ihn selbst bestimmt sind (und in ein anderes Netzsegment übertragen werden müssen).

Unter **Routing** versteht man die Art und Weise, wie Daten(-Pakete) in dezentralen Netzwerken verarbeitet werden. Konkret geht es um die Findung des (richtigen) Weges für die Daten. Z.B. wird unterschieden, ob die Daten im eigenem Netz weitergeleitet werden / verbleiben, ob sie an ein übergeordnetes netz weitergegeben / weitergeleitet werden oder ob sie gar gelöscht werden.

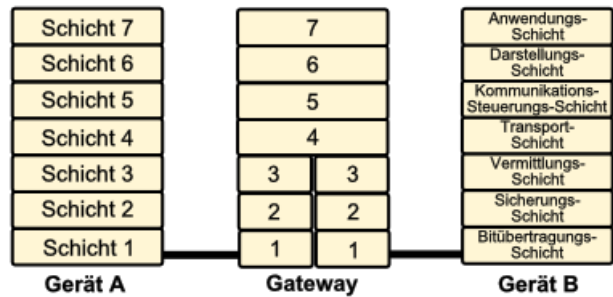
Jeder Router prüft also, ob das Datenpaket im eigenen Netz bleibt oder nach außen weitergeleitet wird. Das Hilfsmittel dafür ist die Routing-Tabelle. In den äußeren Netzwerken besit-

zen die Knoten-Rechner ebenfalls Routing-Tabellen für bekannte Ziel-Punkte oder weitere vermittlungs-Rechner.

Damit die Daten-Pakete nicht unendlich durch Netz irren, werden die Vermittlungs-Versuche gezählt und bei einer bestimmten Anzahl gelöscht.

### 2.5.x.y.z. Gateway

arbeitet ab OSI-Schicht 4 aufwärts (Transport-, Kommunikationssteuerung-, Darstellungs- und Anwendungsschicht)



- 
- 
- 

in lokalen Netzen versteht man unter Gateway den Rechner / die netzwerk-Einheit, welche die Verbindung zum Internet oder einem anderen Netz realisiert (also eher einen Router)

<b>Definition(en): Gateway</b>
Ein Gateway ist eine Netzwerk-Knoten (Hardware, Software oder Kombination aus beidem), der eine Verbindung / Schnittstelle zwischen zwei Netzwerken / unterschiedlichen Netzwerk-Segmenten herstellt und die Daten und Protokolle in geeigneter Form umsetzt.

---

## **Wege-Wahl-Verfahren**

Aufgaben / Ziele:

geringe Übertragungszeiten (geringe Anzahl von genutzten Knoten; kleine Leitungslängen, Ausnutzung der Leitungskapazität, hohe Übertragungs-Geschwindigkeit des Mediums)

geringe Übertragungs-Kosten

gute Auslastung der Kapazitäten (Leitungen, Knoten)

Optimierung des Netzdurchsatzes

Probleme / Konflikte:

soll so einfach, wie möglich, aber so effektiv wie benötigt sein

adaptiv auf bestehenden und veränderliche Topologien / Lasten / ...

Robustheit in Fehlersituationen

Fairness gegenüber jeder Einzelverbindung (Netzgleichheit, )

## **Domain Name Service (DNS)**

übersetzt die technischen Adressen in für Menschen lesbare und verständliche Namen

historisch nur wenige Domains

später immer wieder erweitert

Aufgaben:

Zuordnung IP zu Name

Zuordnung Name zu IP

→ Datei hosts

für Zwecke der speziellen Behandlung von Namen / Adressen in einem Subnetz bzw. für die Station

## **Datenübertragung**

### **Probleme**

Duplikate

doppelt / mehrfach gesendete Pakete, die wegen fehlender oder verspäteter Quittierung nochmals gesendet werden → Erkennung an Sequenz-Nummer → werden ignoriert

Reihenfolge-Fehler

durch unterschiedliche Wegewahl und den damit resultierenden Lauflängen der Signale kommen Pakete ev. in ungeordneter Reihenfolge an → Erkennung an Sequenz-Nummer → werden sortiert

Fehl-Adressierung

durch verfälschter oder fehlerhafter Adresse

---

## **Sicherungs-Verfahren**

Sequenz-Nummer (Pakete / Nachrichten werden fortlaufend durchnummeriert)

Quittierungs-Verfahren

Übertragungsfehler

Verfälschungen durch:

Stör-Signale, Speicherfehler,

→ Nutzung von Error Correcting Codes

Daten-Verlust durch:

Geräte-Ausfälle,

→ Wiederholung der Daten-Übertragung

## Übersicht der verschiedenen Netzwerk-Zwischenknoten

Geräte-Name	Aufgaben / Funktionen	arbeitet auf Schicht ...		Lage bezügl. Collisions-Domäne	?hat IP	? hat MAC	Bandbreite
<b>Repeater</b>	einfache Verstärkung eines Signal's Gerät innerhalb eines Netzes Verstärker	ISO-OSI: 1 (Bit-Übertragung) TCP/IP: 1 (Link)	Gerät unsichtbar für höhere Schichten keine eigene Intelligenz teilt das logische Netz in physikalische Segmente praktisch Bus-Topologie	liegt innerhalb einer CD	nein	nein	unverändert bei passenden Geräten
<b>Medien-Konverter</b>	wandelt Signale einer Technologie in die einer anderen um	ISO-OSI: 1 (od. 2)	praktisch auch Repeater		nein	1	leicht verringert
<b>Hub</b>	verbindet mehrere Host's eines Netzes Gerät innerhalb eines Netzes Verstärker + (dummer) Verteiler	ISO-OSI: 1 (Bit-Übertragung) TCP/IP: 1 (Link)	physikalisch Stern-Topologie praktisch keine Intelligenz praktisch nur eine aktive Verbindung pro Hub teilt das logische Netz in physikalische Segmente erscheint als Bus-Topologie	liegt innerhalb einer CD	nein		alle angeschlossene n Geräte teilen sich die Bandbreite
<b>Bridge</b>	verbindet zwei gleichartige Netze (LAN-Segmente) trennt Gesamt-Netz in Collisions-Domänen Grenz-Übergang Verstärker + Verbinder	ISO-OSI: 2 TCP/IP:	für LAN-Erweiterungen mit unabhängigen CD's kann aber auch Netze mit verschiedenen physikalischen Eigenschaften verbinden (aber gleicher Adressierung) Gerät unsichtbar für höhere Schichten (höhere Schichten müssen aber kompatibel sein) ist Protokoll-transparent unabhängiger Daten-Verkehr in jedem Segment → guter Lasten-Ausgleich selektive Weiterleitung von Paketen (nur die für das spezifizierte andere Netz) geringe Intelligenz (lernt + interpretiert MAC-Adr.) kommunizieren untereinander und verhindern Schleifen	trennt 2 CD's	nein	(mind.) 2	Austausch-Geschwindigkeit wird vom langsameren Netz bestimmt

Geräte-Name	Aufgaben / Funktionen	arbeitet auf Schicht ...	weitere Aufgaben / Eigenschaften / ...	Lage bezügl. Collisions-Domäne	?hat IP	? hat MAC	Bandbreite
<b>Switch</b>	verbindet mehrere Host's eines Netzes selektive Weiterleitung (Netzwerk-Weiche, Verteiler, Umschalter) Verstärker + (intelligenter) Verteiler trennt Gesamt-Netz in Collisions-Domänen	ISO-OSI: 2 (Sicherung) selten: 3 (Vermittlung) TCP/IP: 1 (Link)	intelligente Verteilung der Pakete (Verkehrs-Management) (selten: selektives Paket-Verteilen) untersucht Daten-Pakete nach MAC-Adressen erzeugt pro Port eine eigene Collisions-Domäne sonst wie Bridge	liegt innerhalb einer CD	nein (selten)	1	je Verbindung steht volle Bandbreite zur Verfügung
<b>Router</b>	verbindet zwei verschiedene (autarke / eigenständige) Netze Zuordnung von Paketen zu unterschiedlichen Netzwerken Gerät an der "Außen-Grenze" eines Netzwerk's Verstärker + Verbinder + Vermittler	ISO-OSI: 3 (Vermittlung) TCP/IP: (Internet)	verbindet unterschiedliche Protokolle bis OSI-Schicht 3 Adressierung auf Schicht 3 (z.B. IP) muss gleich sein haben eigenes Betriebssystem meist Multi-Protokoll-fähig gezielte Weiterleitung von Paketen über Routing-Tabelle keine Weiterleitung von Broadcast's ermöglichen Adressierung über das eigenen Netz hinaus selektive Weiterleitung von Paketen (nur die nicht für das eigene Netz sind)	trennt 2 CD's	ja (mind. 2) WLAN-Router (mind. 3)	meist 2	
<b>Gateway</b>	verbindet Host's auf der Anwendungs-Ebene (z.B. für verteiltes Rechnen) Vermittler + (Verstärker + Verbinder)	ISO-OSI: 7 (Anwendung) TCP/IP: (Application)	praktisch ein eigenständiger Host / Server (mit (Server-)Betriebssystem) übernimmt auch Routing-Aufgaben setzt Protokolle direkt ineinander um übersetzt Anwender-Protokolle ineinander viele WLAN-Router übernehmen die notwendigen / nötigsten Funktionen		kann	kann	



## der heimische WLAN-Router – ein kleines Universal-Gerät

gemeint ist hier der klassische WLAN-Router, wie er heute zu Telefon/Internet-Verträgen dazugehört bzw. gebraucht wird  
typisch für den nicht-professionellen / häuslichen Gebrauch

typische Vertreter (Gemein-Bezeichnungen):

- Fritz!-Box
- Speedport
- Kabelmodem

meist mit einem Linux-Betriebssystem ausgestattet

dann i.A. Quellen-offen

kann durch eigenes Betriebssystem (MOD's) ausgetauscht werden

Vorteil der erweiterten Leistungs-Fähigkeit, ev. Garantie-Verlust (bei mir ist erst 1 Router in 20 Jahren kaputt gegangen (und der durfte auch schon mal kaputt gehen))

### **Kombination aus:**

- **Router** ermöglicht die Weiterleitung von entsprechend adressierten Daten-Paketen in ein anderes Netz (hier das Netz des Internet-Provider's (ISP))  
ev. auch das separate WLAN
- **Modem** setzt die Daten-Pakete in die passenden Signal (einschließlich Codierung) für das Netz des ISP um (z.B. ISDN, VDSL, ADSL, Kabel)  
bedient den WAN-Port
- **Switch** ermöglicht den Anschluss mehrerer (meist 4) Host's zu einem Netzwerk  
bedient die LAN-Port's (häufig bis 1 GB-LAN)
- **WLAN-Access Point** stellt ein privates WLAN bereit (kann Erweiterung des geschwichten Netzes sein oder ein eigenständiges Netz bilden)

### **Funktionell kommt ev. hinzu:**

- **Firewall** Überwachung von Verbindungen, TCP/IP-Port's
- **Internet-Telefonie  
VoIP (Voice over IP)** Herstellen von klassischen Telefon-Gesprächen über das Internet
- 

benötigt:

Einwahl-Parameter (Protokoll, Adressen, Optionen, ...) vom ISP

Service-Verbindung (für Web-Bedien-Oberfläche; einschließlich Account mit Name und Password)

selten notwendig:

---

Angaben zum eigenen Netzwerk (wenn nicht DHCP)  
Angaben zum eigenen WLAN

die Anpassungen des Administrator-Account's ist unbedingt zu empfehlen

Anpassung des eigenen LAN (ev. abweichend von den Standard-Netzen 192.168.0.x bzw. 192.168.1.x

Anpassung des WLAN's mit eigener SSID und eigenem Passwort für das WLAN ebenfalls zu empfehlen

MAC-Filter / Zulassen von bekannten Geräten ebenfalls eine wichtige Einstellung für ein sicheres eigenes Netzwerk mit möglichst wenigen (handhabbaren) Verfahren / Veränderungen

---

### Aufgaben:

1. Stellen Sie in einem kleinem freien Vortrag an einem (altem) WLAN-Router oder einem Funktions-Modell die verschiedenen Netzwerk-Funktionen vor!
2. Uns steht ein Heim-Netzwerk mit den folgenden Geräten und Ausstattungen zur Verfügung: An einem WLAN-Router (4 Port-LAN, 1 Port-WAN, 3 WLAN-Antennen, 2 Telefon-Anschlüsse, 1 USB-3.0-Port) sind drei Computer per LAN und ein Laptop per WLAN angeschlossen. Die LAN-Verbindungen sind für 100 MBit/s und das WLAN für 55 MBit/s bei 5 GHz ausgelegt. Der WAN-Anschluss ist mit dem Haus-DSL-Anschluss vom Internet Service Provider (ISP) verbunden. Laut Vertrag steht eine Download-Geschwindigkeit von 33 MBit/s und eine Upload-Geschwindigkeit von 10 MBit/s zur Verfügung (Maximal-Geschwindigkeiten; auch so beobachtet!). Allgemein gilt für alle Verbindungen praktisch nur eine Geschwindigkeit von 80 % - bezogen auf die theoretisch möglichen.
  - a) Stellen Sie das Netzwerk graphisch dar! Notieren Sie die Geräte-Namen / -Typen an die von Ihnen verwendeten Symbole! Geben Sie für jede mögliche Verbindung zwischen zwei Geräten die theoretische(n) Geschwindigkeit(en) an!
  - b) Zwischen zwei Computern soll eine Video-Datei von 152 MByte direkt kopiert werden. Berechnen Sie die Übertragungs-Zeit in Sekunden und Minuten! (Geben Sie immer eine – für Dritte – nachvollziehbare Berechnung an!)
  - c) Berechnen Sie die Übertragungs-Zeiten (s und min) für das Kopieren der Datei auf den Laptop!
  - d) Weil der eine PC keine Freigaben usw. hat, soll die Datei über eine Cloud ausgetauscht werden. Für die Cloud selbst und die Verbindung zu ihr braucht keine extra Verzögerung beachtet werden. Diese ist durch den ISP ausgeschlossen worden. Wie lange dauert die Datei-Übertragung nun?  
für die gehobene Anspruchsebene:
3. Die Video-Datei (Aufgabe 2d) soll verschlüsselt übertragen werden. Dazu wird ein Programm benutzt, dass den Datendurchsatz je Gebrauch um 0,5 s/MBit verzögert. Wie lange braucht die Übertragung (von unverschlüsselt bis unverschlüsselt) nun? Erklären Sie Ihre Berechnung dem Kurs!
4. Einer der PC's wird nach dem Geburtstag gegen einen Laptop ausgetauscht. Wie lange muss man nun warten, bis die Datei auf dem neuen Laptop über die Cloud zur Verfügung steht?

---

## 4.2. Simulation von Netzen mit Filius

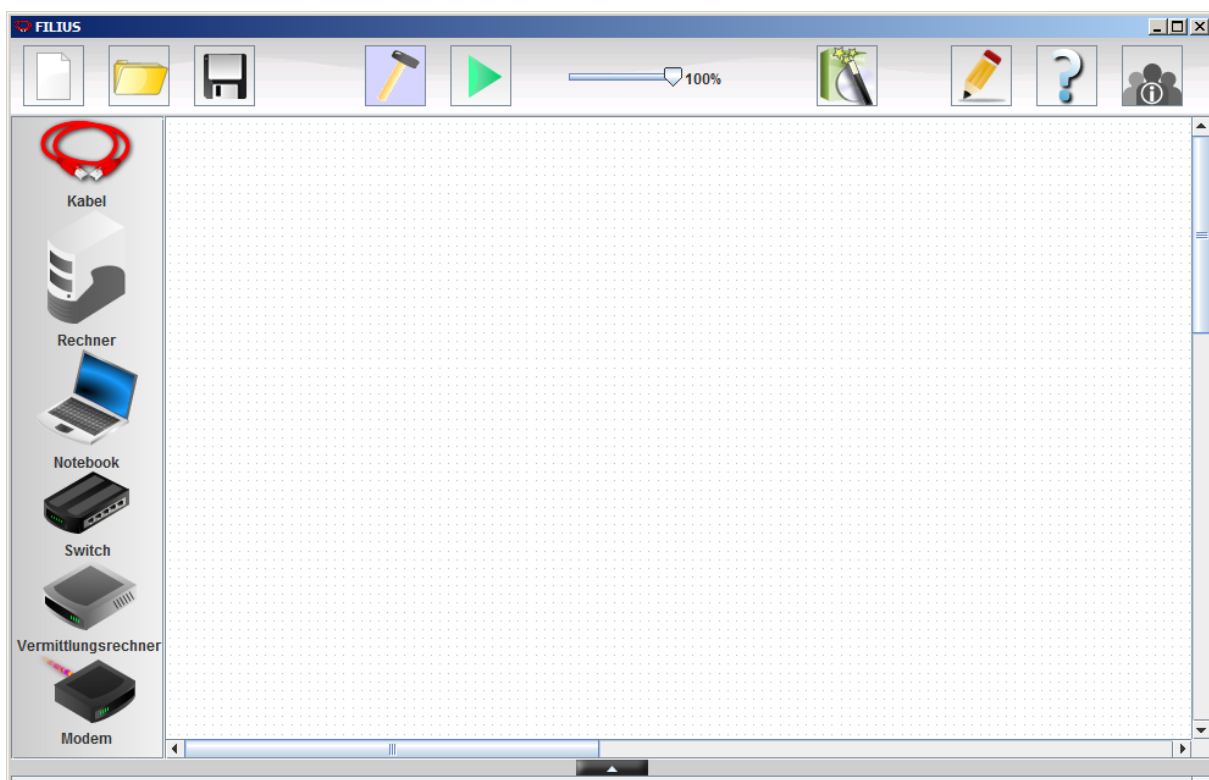
Filius ist eine Software, die es uns gestattet virtuelle Netzwerke aufzubauen, zu konfigurieren und auszuprobieren. Dabei sind vor allem die "versteckten" Informationen für uns interessant, um die Hintergründe und Abläufe besser kennen zu lernen.

FILIUS ... Freie Interaktive Lernumgebung für Internetworking der Universität Siegen

### 4.2.0. Wege zu Filius

Start vom IoStick

eigene Installation



Die individuellen Einstellungen werden im Benutzer-Ordner .filius gespeichert. Durch Löschen dieses Ordner's gelangt man wieder zu den Grund-Einstellungen. Das lässt sich z.B. durch ein einfaches BAT-Programm realisieren, was z.B. immer bei einer Nutzer-Anmeldung gestartet wird (z.B. auch für Leistungskontrollen oder Prüfungen).

## 4.2.1. Aufbau von Netzen

Nach dem Start von Filius sehen wir meist das letzte bearbeitete Projekt. Mit [ Strg ] + [ N ] erstellen wir uns ein neues (leeres) Projekt. Mangels Menü lässt sich auch die nebenstehende Schaltfläche benutzen.

Die Schaltflächen "Öffnen" und "Speichern" erklären sich von alleine. Eine Funktion "Speichern unter ..." gibt es nicht. Bei jedem "Speichern" wird der Speichern-unter-Dialog aufgerufen. Beim einfachen Bestätigen wird die benutzte Datei gespeichert (überschrieben).

Die Speichern-unter-Funktion bekommen wir durch Ändern des Datei-Namens oder des Speicher-Ortes. Vor allem, wenn man sich also neue Filius-Dateien (\*.FLS) anlegen will, muss man immer gut auf die Benennung achten, sonst ist schnell mal eine andere überschrieben.

Besonders wer Meilensteine seiner Arbeit dokumentieren will, ist hier gefordert.

Die möglichen Grund-Geräte (Netzwerk-Hardware) liegen in der linken Werkzeug-Leiste bereit. Ein entsprechendes Gerät wird durch Ziehen auf die Arbeitsfläche positioniert.



Mit Kabeln verbinden wir die Geräte. Dazu müssen immer nacheinander die beiden Geräte angeklickt werden. Das Entfernen eines Kabels ist (nur) über einen Rechts-Klick auf das Kabel möglich.

(Für die technischen Freak's: Ob es sich um "Patch"- oder "Cross over"-Kabel handelt, wird hier ignoriert / übergangen. Es ist immer das "richtige" Kabel bzw. ein moderner erkennender Netzwerk-Port am Endgerät.)

Ein "Rechner" im Sinne von Filius ist ein Server – also ein Rechner, der eine Leistung im Netz zur Verfügung stellt. Auch wenn der Rechner scheinbar keinen Monitor hat, können wir Software installieren und auch beobachten.

Das "Notebook" ist ein klassischer PC im Netz. Ob dieser als Client (Klient) oder als Peer (Partner) fungiert, wird durch die anderen Netzwerk-Geräte und die "installierte" Siftware bestimmt.

"Switch"e dienen zum Aufbau einer Stern-förmigen Netzwerk-Struktur und damit dem Verbinden mehrerer Rechner.

Was hier allgemein "Vermittlungsrechner" genannt wird, sind praktisch die Router bzw. Bridge's. Wann wir statt eines Switches einen Vermittlungsrechner brauchen klären wir später noch genau.

Das "Modem" ist seit je her der Inbegriff des Verbindungs-Gerätes zum Internet. Hier sind Modem's zur Verbindung "weiter" entfernter oder "lokaler" Netze gedacht. Auch diese Geräte-Klasse sehen wir uns noch genauer an.

Normalerweise befinden wir uns zu Anfang im "Entwicklungsmodus" [ Strg ] + [ D ]. Hier wird das Netz zusammengestellt und die wichtigen Vorgaben, wie Benennungen und Adressierungen vorgenommen.



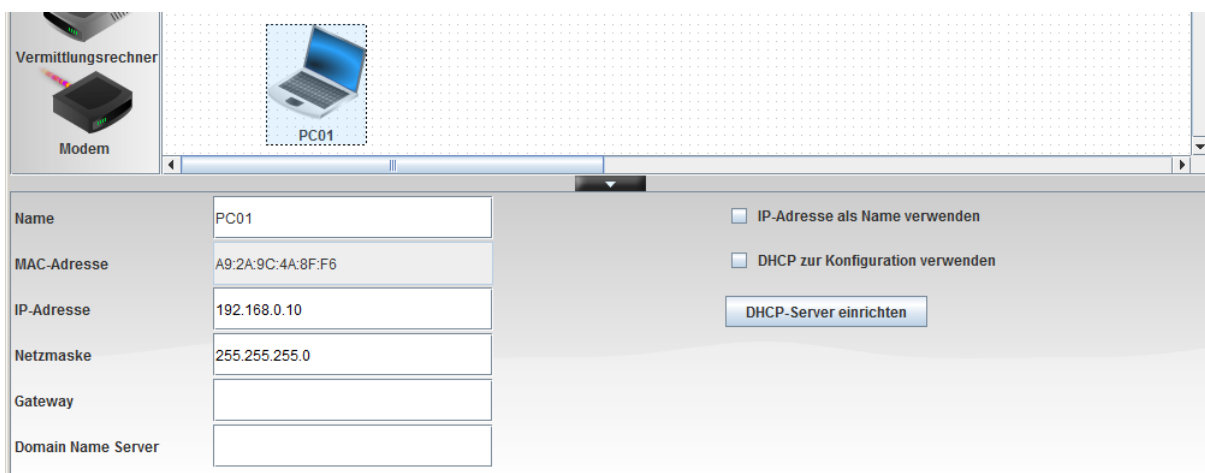
Der Aktionsmodus [ Strg ] + [ R ] ist dann zum praktischen Ausprobieren. Er schaltet die Geräte sozusagen an und ermöglicht die Beobachtung der Netzwerk- und Geräte-Aktivitäten.

Bleibt noch der Dokumentations-Modus, der uns z.B. die Beschreibung unseres virtuellen Netzes ermöglicht.



### 4.2.1.1. Einrichten und Nutzen von Netzendgeräten in Filius

Nachdem man z.B. ein Gerät auf der Arbeitsfläche positioniert hat, zeigt sich unten im Programm der Detail- oder Eigenschaften-Bereich. Hier können wir z.B. für ein Notebook einen Namen und die Netzwerk-Adressen festlegen.



Die MAC-Adresse ist vom "Hersteller" fest vergeben worden. Die IP-Adressen können wir relativ frei belegen.

Als nächstes starten wir den PC durch ein Umschalten in den Aktivitäts-Modus. Jetzt sehen wir den Bildschirm mit dem Desktop. Die einige Applikation ist derzeit der Assistent für die Software-Installation.



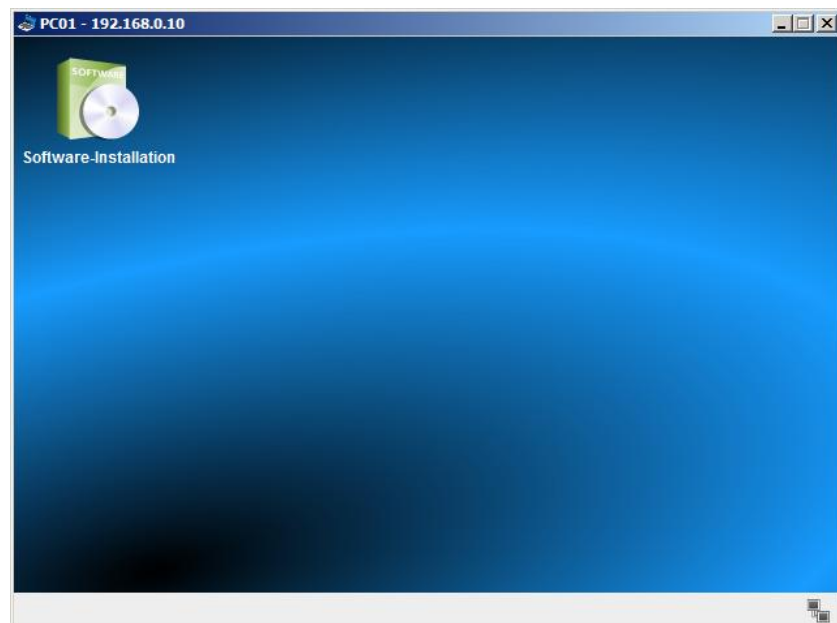
Ohne installierte Software kann unser PC praktisch nicht viel.

Wir wollen aber kommunizieren und die verschiedenen Netzwerk-Protokolle ausprobieren.

Durch Filius werden die wichtigsten Protokolle für Netzwerke sehr schön abgedeckt.

Alle relevanten Abläufe und Probleme können so aufgedeckt werden.

Für weiterführende Netzwerk-Erkundungen muss man dann auch zu professionellen Programmen greifen.



Die gewünschte Software wird aus der rechten Liste durch Doppelklick oder über den grünen Pfeil in die "Installiert"-Liste befördert.

Nach dem "Änderungen übernehmen" stehen die Applikationen installiert auf dem Desktop bereit.

Zuerst genügt uns mal die "Befehlszeile". Dies entspricht einer Konsole oder auch einem Terminal. In älteren Windows-Versionen sprach man auch von der "MS-DOS Eingabeaufforderung".

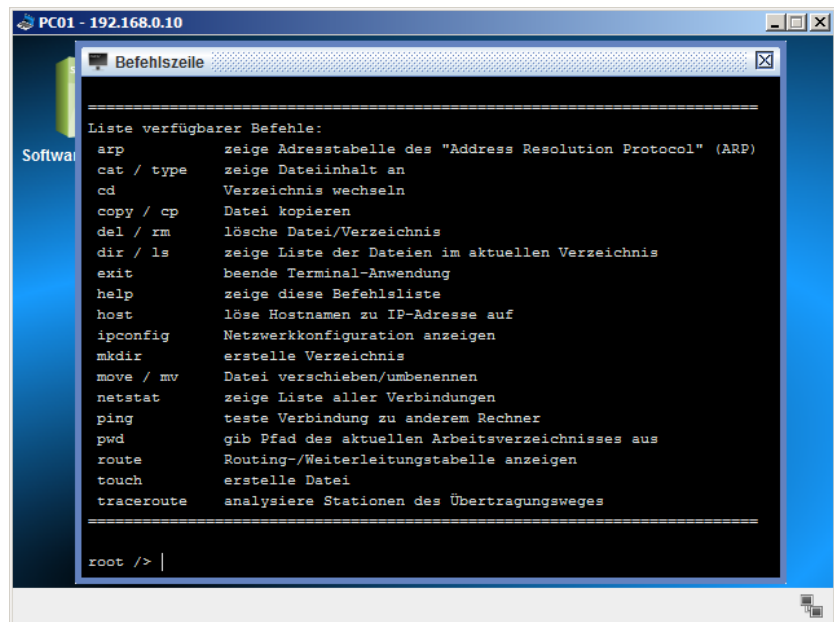
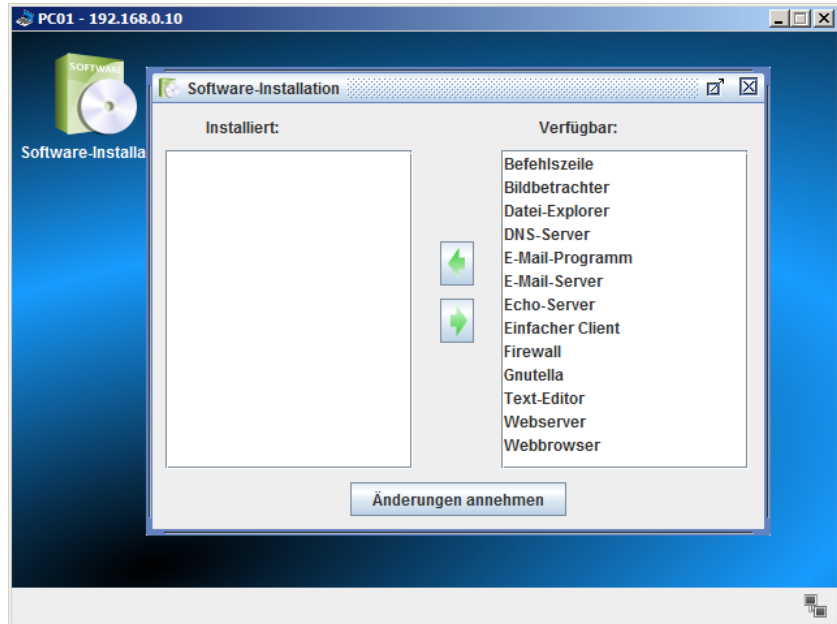
Die Befehlszeile ermöglicht die Nutzung von Kommandozeilen- bzw. Konsolen-Befehlen.

Die Auswahl der Befehle ist auf eine kleine Gruppe eingeschränkt.

Viele Befehle kennt man vielleicht von speziellen Aktivitäten am eigenen PC. Die Konsole wird heute immer mehr nur noch von System-Betreuern benutzt. Sie ist und bleibt – auch auf dem eigenen PC – eine der sichersten Bedienmöglichkeiten. Da ist kein graphisches System, das abstürzen kann, weil der Nutzer zu hektisch geklickt hat.

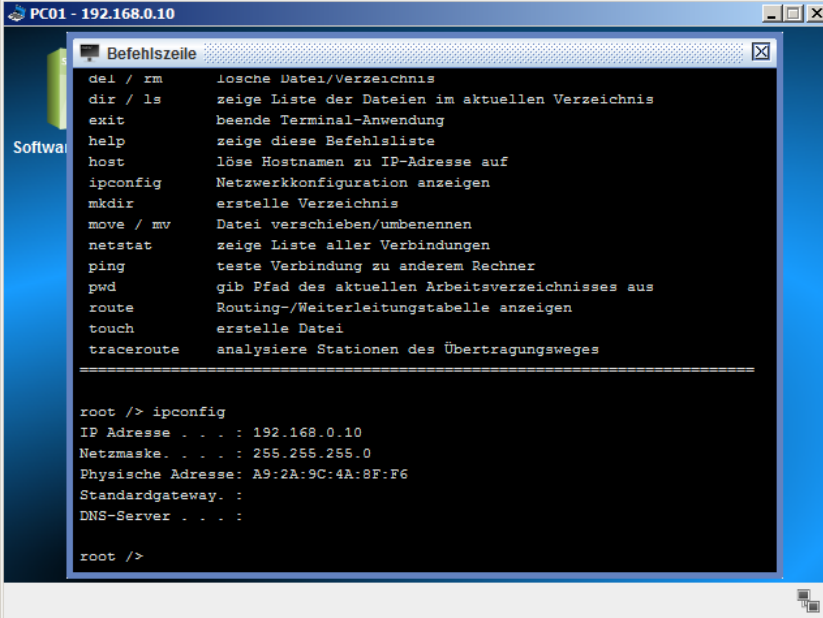
Wenn man die Liste der verfügbaren Befehle später nochmal braucht, dann ist diese über **help** zu erreichen.

Die Konsole sollte man immer ordnungsgemäß über **exit** verlassen.



Der Befehl **ipconfig** zeigt uns die Einstellungen der Netzwerkschnittstelle (hier z.B. die LAN-Karte) an.

Der Befehl ist auch in den Konsolen von Windows und verfügbar. Deren Optionen lassen sich dort mittels `/?` bzw. `-h` abrufen.



```
PC01 - 192.168.0.10
Befehlszeile
del / rm      lösche Datei/Verzeichnis
dir / ls     zeige Liste der Dateien im aktuellen Verzeichnis
exit        beende Terminal-Anwendung
help        zeige diese Befehlsliste
host        löse Hostnamen zu IP-Adresse auf
ipconfig     Netzwerkkonfiguration anzeigen
mkdir       erstelle Verzeichnis
move / mv    Datei verschieben/umbenennen
netstat     zeige Liste aller Verbindungen
ping        teste Verbindung zu anderem Rechner
pwd         gib Pfad des aktuellen Arbeitsverzeichnisses aus
route       Routing-/Weiterleitungstabelle anzeigen
touch       erstelle Datei
tracert     analysiere Stationen des Übertragungsweges

=====

root /> ipconfig
IP Adresse . . . : 192.168.0.10
Netzmaske . . . : 255.255.255.0
Physische Adresse: A9:2A:9C:4A:8F:F6
Standardgateway. :
DNS-Server . . . :
```

Der **ping**-Befehl ermöglicht das Prüfen einer Netzwerk-Verbindung. Dabei geht es zum Einen um das Vorhandensein einer Netzwerk-Ressource und zum Anderen um die Übertragungs-Geschwindigkeit. Es werden beim Ping mehrere Daten-Pakete gesendet. In der Konsole sehen wir dann, ob das Paket erfolgreich versendet wurde und welche Zeit dafür gebraucht wurde. Diese Zeiten sollten innerhalb eines lokalen Netzes immer unter oder im Millisekunden-Bereich liegen. Der angepingt Rechner antwortet mit einem pong. Dieses Kommando realisiert das Zurücksenden der verschickten Pakete. Ein Ping ist also immer das hin- und herschicken von Daten- Paketen. Bei der Berechnung der eigentlichen Geschwindigkeit muss also beachtet werden, das der Weg doppelt zurückgelegt wird.

Kommen die vier Daten-Pakete nicht alle an – und dies auch bei einem 2. oder 3. Test – dann liegt ein technisches Problem vor. Meist sind das defekte Kabel oder Netzwerk-Geräte. Da wir im Augenblick nur ein Gerät in unserem virtuellen System verfügbar, können wir natürlich nur dieses Gerät testen. Man kann immer auch seine eigene IP-Adresse mit einem ping testen. Dadurch weiss man, das die eigene Netzwerk-Karte funktioniert und eine gültige IP-Adresse hat.

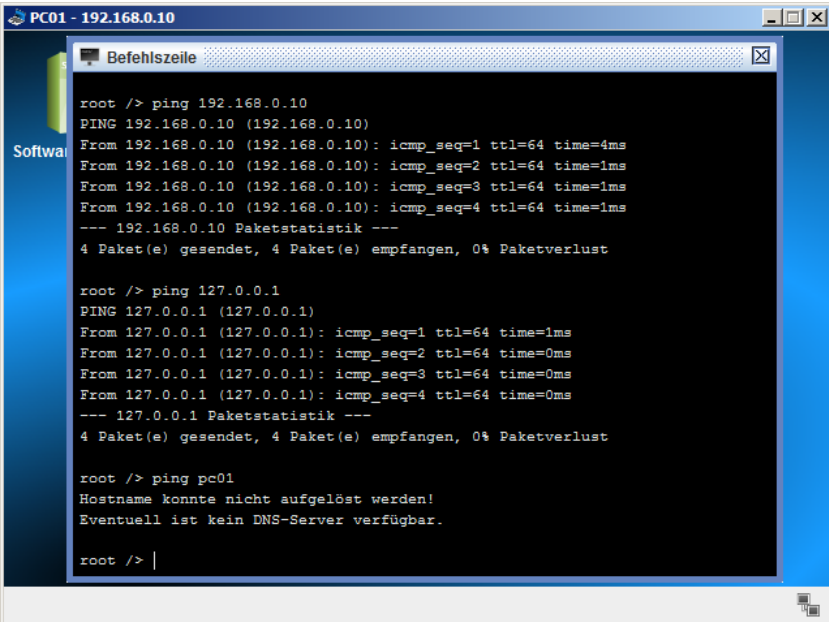
### ping eigene\_IP

Für eigene\_IP setzen wir die eingerichtete oder mit ipconfig ermittelte IPv4 ein.

### ping loopback\_IP

Die klassische loopback\_IP ist **127.0.0.1**. Sie sollte immer funktionieren, wenn das Netzwerk mit TCP/IP und dem IPv4-Adressen arbeitet.

(Praktisch sind alle Adressen von 127.0.0.1 bis 127.255.255.254 als Loopback-Host's vorgesehen.) Bei IPv6 ist die Loopback-Adresse die `::1`.



```
PC01 - 192.168.0.10
Befehlszeile

root /> ping 192.168.0.10
PING 192.168.0.10 (192.168.0.10)
From 192.168.0.10 (192.168.0.10): icmp_seq=1 ttl=64 time=4ms
From 192.168.0.10 (192.168.0.10): icmp_seq=2 ttl=64 time=1ms
From 192.168.0.10 (192.168.0.10): icmp_seq=3 ttl=64 time=1ms
From 192.168.0.10 (192.168.0.10): icmp_seq=4 ttl=64 time=1ms
--- 192.168.0.10 Paketstatistik ---
 4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root /> ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1)
From 127.0.0.1 (127.0.0.1): icmp_seq=1 ttl=64 time=1ms
From 127.0.0.1 (127.0.0.1): icmp_seq=2 ttl=64 time=0ms
From 127.0.0.1 (127.0.0.1): icmp_seq=3 ttl=64 time=0ms
From 127.0.0.1 (127.0.0.1): icmp_seq=4 ttl=64 time=0ms
--- 127.0.0.1 Paketstatistik ---
 4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root /> ping pc01
Hostname konnte nicht aufgelöst werden!
Eventuell ist kein DNS-Server verfügbar.

root /> |
```



---

In den Konsolen echter PC's können wir auch:

**ping *PC\_Name***

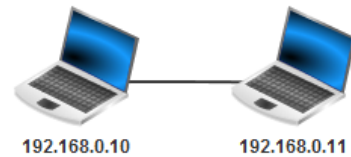
benutzen. In unserem virtuellen System funktioniert das nicht, weil es noch keinen Dienst gibt, der die PC-Namen in IP-Adressen umwandelt oder umgekehrt. Das erledigt der DNS-Dienst (→). Für die Loopback-Adresse kann man dann auch:

**ping localhost**

verwenden.

## 4.2.1.2. Verbinden von Netzendgeräten in Filius

Das Kabel-Ziehen zum Verbinden von Geräten wird von den meisten Nutzern ganz intuitiv vorgenommen. Unter Verwendung des Kabel-Werkzeug's klicken wir immer zuerst auf die 1. Endstelle und dann auf die 2. Der Mauszeiger gibt auch ein entsprechendes Feedback.



Den Kabel-Modus verlässt man mit [ ESC ].

Als Verbindungs-Test kennen wir den ping-Befehl, den wir nun auf einem Rechner ausprobieren.

Der Rechner 192.168.0.10 wird als Quelle genutzt. Zur Sicherheit pingen wir den eigenen Rechner zuerst an und dann den eigentlichen Zielrechner.

Wie man unschwer erkennen kann, funktionieren beide Netzwerk-Schnittstellen und die Verbindung. Kein Paket ist verloren gegangen und die Antwortzeiten sind ebenfalls recht stabil.

Auffällig ist lediglich die doppelt so große Zeit für den ersten Ping. Dieses Phänomen - welches wir immer bei neuen Verbindungen beobachten werden – erklären wir gleich. Das "Problem" ist hier das ARP-Protokoll (→ [ARP](#)).

Eine Wiederholung des Ping-Befehls bestätigt den Effekt. Jetzt sind die Zeiten recht stabil bei 100 ms.

```
192.168.0.10
Befehlszeile
route Routing-/Weiterleitungstabelle anzeigen
touch erstelle Datei
tracertool analysiere Stationen des Übertragungsweges

root /> ping 192.168.0.10
PING 192.168.0.10 (192.168.0.10)
From 192.168.0.10 (192.168.0.10): icmp_seq=1 ttl=64 time=0ms
From 192.168.0.10 (192.168.0.10): icmp_seq=2 ttl=64 time=0ms
From 192.168.0.10 (192.168.0.10): icmp_seq=3 ttl=64 time=0ms
From 192.168.0.10 (192.168.0.10): icmp_seq=4 ttl=64 time=0ms
--- 192.168.0.10 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root /> ping 192.168.0.11
PING 192.168.0.11 (192.168.0.11)
From 192.168.0.11 (192.168.0.11): icmp_seq=1 ttl=64 time=202ms
From 192.168.0.11 (192.168.0.11): icmp_seq=2 ttl=64 time=100ms
From 192.168.0.11 (192.168.0.11): icmp_seq=3 ttl=64 time=100ms
From 192.168.0.11 (192.168.0.11): icmp_seq=4 ttl=64 time=100ms
--- 192.168.0.11 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root />
```

```
192.168.0.10
Befehlszeile
From 192.168.0.10 (192.168.0.10): icmp_seq=3 ttl=64 time=0ms
From 192.168.0.10 (192.168.0.10): icmp_seq=4 ttl=64 time=0ms
--- 192.168.0.10 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root /> ping 192.168.0.11
PING 192.168.0.11 (192.168.0.11)
From 192.168.0.11 (192.168.0.11): icmp_seq=1 ttl=64 time=202ms
From 192.168.0.11 (192.168.0.11): icmp_seq=2 ttl=64 time=100ms
From 192.168.0.11 (192.168.0.11): icmp_seq=3 ttl=64 time=100ms
From 192.168.0.11 (192.168.0.11): icmp_seq=4 ttl=64 time=100ms
--- 192.168.0.11 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root /> ping 192.168.0.11
PING 192.168.0.11 (192.168.0.11)
From 192.168.0.11 (192.168.0.11): icmp_seq=1 ttl=64 time=102ms
From 192.168.0.11 (192.168.0.11): icmp_seq=2 ttl=64 time=100ms
From 192.168.0.11 (192.168.0.11): icmp_seq=3 ttl=64 time=102ms
From 192.168.0.11 (192.168.0.11): icmp_seq=4 ttl=64 time=100ms
--- 192.168.0.11 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

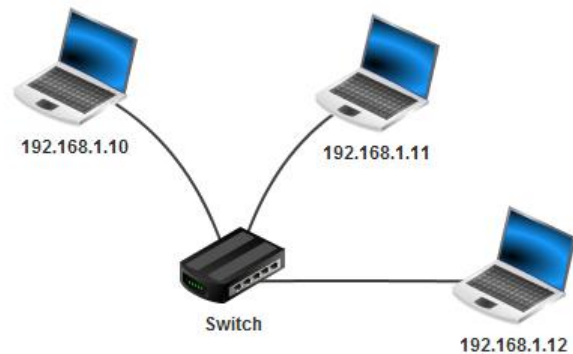
root /> |
```

### Aufgaben:

- 1.
2. *Macht es Sinn, eine Verbindung jeweils von beiden Rechnern zu testen, um eine bessere Aussage zu bekommen? Begründen Sie Ihre Meinung!*
- 3.

Wir stoßen beim Kabelziehen aber schnell an Grenzen, weil die Endgeräte (Notebook's und Server) nur jeweils über eine Netzwerk-Schnittstelle verfügen – da ist die maximale Anzahl angeschlossener Geräte schnell erreicht.

Bei den Netz-Topologien (→ [Topologie: Struktur-Aspekt](#)) haben wir schon verschiedene Anschluß-Szenarien kennen gelernt. Die in lokalen Netzen vorrangig verwendete Topologie ist der Stern (→ [Stern-Topologie](#)). Als Zentral-Gerät benötigen wir einen Switch.



Nun interessiert natürlich, ob das Netz gleich nach dem Dazwischen-Schalten funktioniert, oder ob wir noch Einstellungen am Switch vornehmen müssen? Braucht der z.B. eine eigene IP?

Also testen wir mit den uns bekannten Konsolen-Programmen:

Offensichtlich funktioniert ein **Anpingen** der anderen Station ohne Probleme – das Netz mit Switch funktioniert.

Schauen wir uns die Eigenschaften des Switches an, dann finden wir auch gar keine Möglichkeit, eine IP-Adresse festzulegen.

```
192.168.1.10
Befehlszeile
exit      beende Terminal-Anwendung
help      zeige diese Befehlsliste
host      löse Hostnamen zu IP-Adresse auf
ipconfig  Netzwerkkonfiguration anzeigen
mkdir     erstelle Verzeichnis
move / mv Datei verschieben/umbenennen
netstat   zeige Liste aller Verbindungen
ping      teste Verbindung zu anderem Rechner
pwd       gib Pfad des aktuellen Arbeitsverzeichnisses aus
route     Routing-/Weiterleitungstabelle anzeigen
touch     erstelle Datei
traceroute analysiere Stationen des Übertragungsweges

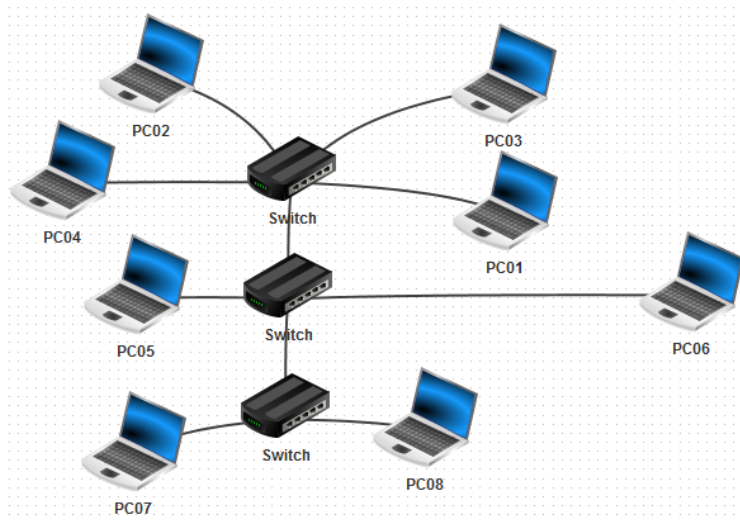
=====

root /> ping 192.168.1.11
PING 192.168.1.11 (192.168.1.11)
From 192.168.1.11 (192.168.1.11): icmp_seq=1 ttl=64 time=404ms
From 192.168.1.11 (192.168.1.11): icmp_seq=2 ttl=64 time=201ms
From 192.168.1.11 (192.168.1.11): icmp_seq=3 ttl=64 time=200ms
From 192.168.1.11 (192.168.1.11): icmp_seq=4 ttl=64 time=200ms
--- 192.168.1.11 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root />
```

### Aufgaben:

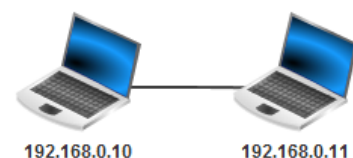
1. Erstellen Sie sich ein virtuelles Netz in Filius mit zuerst nur einem oder mehreren unverbundenen Notebook's (wie oben)!
2. Testen Sie die eigene Netzwerk-Einrichtung und die Funktionsfähigkeit der Geräte-Netzwerkkarte! Wie lange braucht ein Ping-Paket eigentlich auf der eigenen Netzwerkkarte? Erläutern Sie!
3. Testen Sie die Befehle `ipconfig` und `ping` auch auf dem echten Rechner, an dem Sie arbeiten! Probieren Sie nun auch die `ping`-Variante mit dem PC-Namen aus!
4. Wenn Sie nur ein Notebook in Ihrem Filius-Netzwerk haben, dann ergänzen Sie nun ein zweites! Konfigurieren Sie die Adressen so, dass beide Geräte im gleichen Netzwerk sind!
5. Pingen Sie jetzt die Geräte gegenseitig an! Berechnen Sie die durchschnittliche Antwortzeit über alle Ping's! Erklären Sie die angezeigten Antwortzeiten!
6. Bauen Sie nun ein Stern-förmiges Netz mit 4 PC's und einem Switch! Erklären Sie, warum man nun ein solches Zusatzgerät benötigt!
7. Testen Sie wieder die Konnektivität und erklären Sie die Messwerte!
8. Überlegen Sie sich für das angebildete Netzwerk, welche Antwortzeiten für Ping's zwischen PC01 und PC08 zu erwarten sind! Begründen Sie Ihre Voraussage!
9. Vergleichen Sie die erwarteten Ping-Zeiten für die Verbindungen PC01-PC05 und PC01-PC06! Begründen Sie!



Was genau passiert, ist eine klassische Protokoll-Geschichte. Leider zeigt Filius nur Kommunikationen zwischen Rechnern an. Deshalb bringt ein lokaler Ping auch kein Aktivitäts-Protokoll hervor.

Das Aktivitäts-Protokoll kann man sich immer über die rechte Maus-Taste (Kontext-Menü) zum entsprechenden Gerät anzeigen lassen. Gemeint ist dann immer die aktive Verbindung zum nächsten Netzwerk-Gerät.

Sind dagegen zwei PC's mittels Kabel verbunden, dann läßt sich der zweite PC anpingen und wir erhalten ein Aktivitäts-Protokoll.



```

192.168.0.10
Befehlszeile
exit      beende Terminal-Anwendung
help     zeige diese Befehlsliste
host     löse Hostnamen zu IP-Adresse auf
ipconfig Netzwerkkonfiguration anzeigen
mkdir    erstelle Verzeichnis
move / mv Datei verschieben/umbenennen
netstat  zeige Liste aller Verbindungen
ping     teste Verbindung zu anderem Rechner
pwd      gib Pfad des aktuellen Arbeitsverzeichnisses aus
route    Routing-/Weiterleitungstabelle anzeigen
touch    erstelle Datei
tracert  analysiere Stationen des Übertragungsweges

-----

root /> ping 192.168.0.11
PING 192.168.0.11 (192.168.0.11):
From 192.168.0.11 (192.168.0.11): icmp_seq=1 ttl=64 time=390ms
From 192.168.0.11 (192.168.0.11): icmp_seq=2 ttl=64 time=104ms
From 192.168.0.11 (192.168.0.11): icmp_seq=3 ttl=64 time=100ms
From 192.168.0.11 (192.168.0.11): icmp_seq=4 ttl=64 time=102ms
--- 192.168.0.11 Paketstatistik ---
 4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust

root />

```

Der Sender schickt die oben schon beschriebenen vier Pakete los und der Empfänger antwortet mit vier pongs.  
Im Aktivitäts-Protokoll (von Filius) sieht das dann so aus:

### Aktivitäts-Protokoll des TCP/IP-Stack's

Nr.	Zeit	Quelle	Ziel	Protokoll	Schicht	Bemerkungen
1	20:29:11.600	192.168.0.10	192.168.0.11	ARP	Vermittlung	Suche nach MAC für 192.168.0.11, 192.168.0.10: 37:2C:6E:C7:00:4E
2	20:29:11.886	192.168.0.11	192.168.0.10	ARP	Vermittlung	192.168.0.11: E5:B3:E1:CF:E8:6E
3	20:29:11.888	192.168.0.10	192.168.0.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 1
4	20:29:11.989	192.168.0.11	192.168.0.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 1
5	20:29:12.800	192.168.0.10	192.168.0.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 2
6	20:29:12.902	192.168.0.11	192.168.0.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 2
7	20:29:14.000	192.168.0.10	192.168.0.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 3
8	20:29:14.100	192.168.0.11	192.168.0.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 3
9	20:29:15.201	192.168.0.10	192.168.0.11	ICMP	Vermittlung	ICMP Echo Request (ping), TTL: 64, Seq.-Nr.: 4
10	20:29:15.302	192.168.0.11	192.168.0.10	ICMP	Vermittlung	ICMP Echo Reply (pong), TTL: 64, Seq.-Nr.: 4

Ein Aktivitäts-Protokoll erhält man, wenn man im Simulations-Modus mit der rechten Maustaste z.B. auf einen Rechner klickt und dann die Funktion "Datenaustausch anzeigen (...)" auswählt.

In der ersten Zeile ist der ARP-Request zu sehen. Gesucht wird die IP-Adresse 192.168.0.11. Eine zugehörige MAC-Adresse ist noch nicht bekannt. (Hier wird ja beim Request die Broadcast-Adresse FF:FF:FF:FF:FF:FF eingesetzt, was wir hier nicht sehen.)

Die ARP-Antwort sehen wir in Zeile 2. Es wird die zur angefragten IP zugehörige MAC-Adresse geliefert.

Nun kann die eigentliche ping-pong-Kommunikation laufen. Dies gehört zum Internet Control Message Protocol (ICMP). Das ICMP gehört faktisch zum IPv4-Protokoll dazu, wird aber vielfach wie ein eigenes Protokoll behandelt. I.A. wird davon ausgegangen, dass jedes IP-Gerät auch das ICMP versteht.

Im DoD-Modell bleiben wir auf der untersten Schicht. Das kann man gut an der grünen Färbung der Aktivitäten erkennen. Andere Schichten haben andere Farben. Dazu später mehr.

### ICMP-Paket-Typen (Auswahl)

Typ-Nr.	Paket-Typ	Bemerkungen
0	Echo: Antwort	Anfrage: 8
8	Echo: Anfrage	Antwort: 0
9	Router-Angebot	
10	Router-Anwerbung	
11	Zeitüberschreitung	
13	Zeitstempel	Absender
14	Zeitstempel: Antwort	

Viele der ursprünglichen ICMP-Paket-Typen sind mittlerweile abgeschafft. Sie spielen in modernen Netzen keine Rolle mehr. Andere sind durch das ICMPv6 für die neuen Netze ersetzt worden.

Ein Programm, das aktiv mit dem ICMP arbeitet ist traceroute (in Windows: tracert). Mit traceroute lässt sich der Weg einer Anfrage (also der Weg zu einem bestimmten Ziel) anzeigen. Das Programm sendet eine ICMP-Echo-Anfrage an den Ziel-Rechner. Im ICMP-Paket wird der TTL-Wert auf 1 gesetzt. Der erste nachfolgende Rechner prüft die Ziel-Adresse und stellt fest, dass er nicht das Ziel des Paketes ist. Normalerweise würde er das Paket nun weiterleiten. Er setzt TTL um 1 runter. Dadurch wird TTL = 0. Somit wird das Paket ungültig und verworfen und an den Sende-Rechner ein ICMP-Antwort-Paket (Typ 11) gesendet. Der antwortende Rechner trägt noch einen Zeitstempel mit ein. Diese Zeitangabe wird dann von traceroute für die Berechnung der Paket-Laufzeit verwendet und angezeigt.

Nun wiederholt traceroute das Pozedere mit einer um 1 erhöhten TTL. Dadurch wird das Paket einen Internet-Knoten weiter geleitet. An dieser Stelle wird dann TTL = 0 und das Paket wird hier verworfen und ein ICMP-Antwort-Paket gesendet. Die Antwort enthält wiederum einen Zeitstempel für die Laufzeitberechnung.

Traceroute wiederholt nun dieses Verfahren solange, bis das Ziel reicht wird. Zwei Beispiel-Routen sind nebenstehend zu sehen. (Woher weiss traceroute eigentlich die IP's meiner Ziele?)

```

C:\Windows\system32\cmd.exe
C:\Users\dreus>tracert de.wikipedia.org
Routenverfolgung zu de.wikipedia.org [91.198.174.192] über maximal 30 Abschnitte:

 1  <1 ms  <1 ms  <1 ms  fritz.box [192.168.100.2]
 2  23 ms  6 ms   6 ms   62.155.241.33
 3  21 ms  20 ms  20 ms  217.239.54.186
 4  20 ms  25 ms  21 ms  ae2-cr2-esans.wikimedia.org [80.249.209.176]
 5  23 ms  22 ms  22 ms  text-lb.esans.wikimedia.org [91.198.174.192]

Ablaufverfolgung beendet.
C:\Users\dreus>tracert google.de
Routenverfolgung zu google.de [172.217.18.163] über maximal 30 Abschnitte:

 1  <1 ms  <1 ms  <1 ms  fritz.box [192.168.100.2]
 2  7 ms   7 ms   6 ms   62.155.241.33
 3  16 ms  17 ms  16 ms  217.239.42.130
 4  16 ms  16 ms  16 ms  80.156.160.118
 5  *      *      *      Zeitüberschreitung der Anforderung.
 6  19 ms  18 ms  18 ms  74.125.37.124
 7  16 ms  16 ms  16 ms  74.125.37.197
 8  16 ms  16 ms  16 ms  fra15s29-in-f3.1e100.net [172.217.18.163]

Ablaufverfolgung beendet.
C:\Users\dreus>

```

Es existieren auch traceroute-Programme mit graphischer Anzeige der Paket-Routen (z.B.: Open Visual Traceroute). Das ist natürlich viel informativer, weil auch die unbekanntenen Zwischenstationen nun erhellt werden. Einen Eindruck vermittelt die folgende Abbildung.

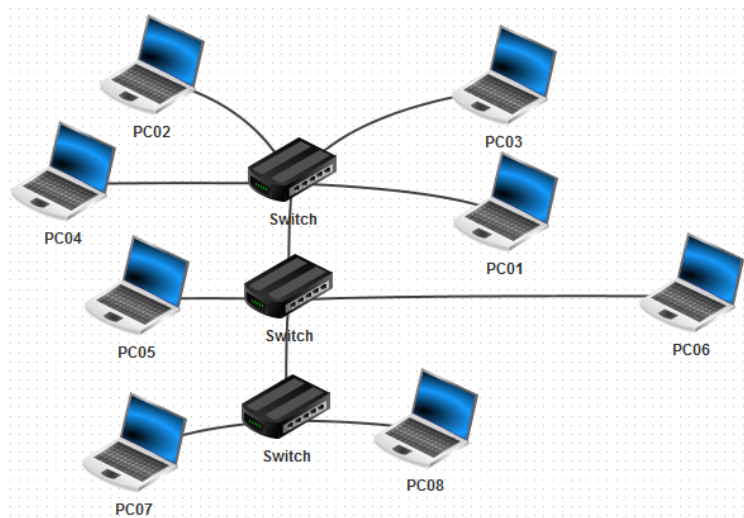
### Aufgaben:

1. Bauen Sie sich in Filius ein Netz aus 2 Notebook's (IP's 192.168.0.10 + 11/24)! Installieren Sie auf dem ersten Notebook eine Befehlszeile! Arbeiten Sie dann die drei folgenden Befehle ab:

```
arp          ping 192.168.0.11          arp
```

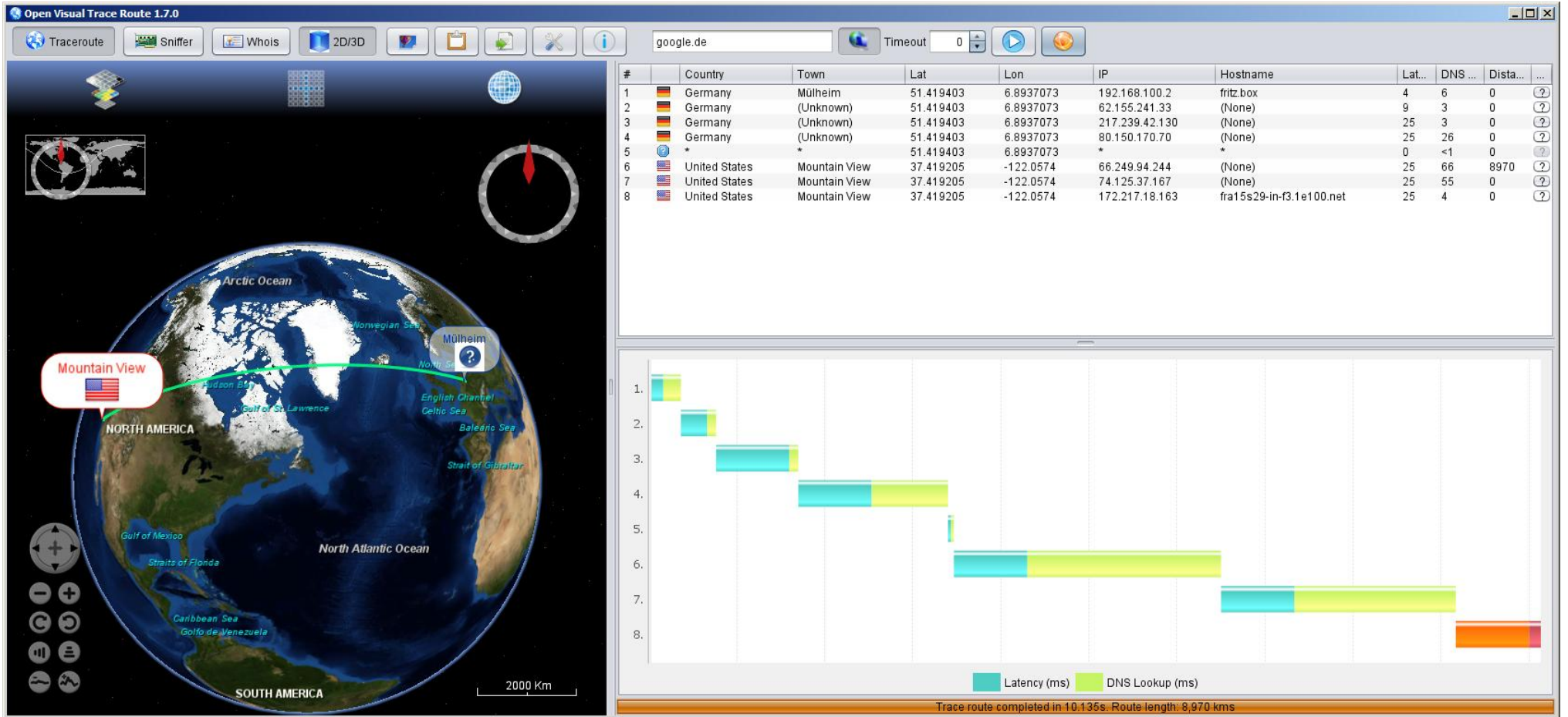
Dokumentieren Sie die Anzeige als Bildschirm-Ausdruck und erläutern Sie, was die Anzeigen in der Befehlszeile bedeuten!

2. Ermitteln Sie auf Ihrem Arbeits-PC die Internet-Route für den Aufruf Ihrer Schul-Homepage!
3. Wählen Sie sich vier Internet-Adressen aus, die sehr wahrscheinlich geographisch weit entfernt gehostet werden (z.B. möglich: Universitäten auf verschiedenen Kontinenten)! Ermitteln Sie die Internet-Routen für die verschiedenen Seiten! Vergleichen Sie die Routen! Erklären Sie Gemeinsamkeiten und Unterschiede!
4. Öffnen Sie das Netz mit den 8 PC's und den 3 Switchen (oder erstellen Sie sich ein solches Netz) und führen Sie ein traceroute für das loop-back am Rechner PC01 aus! Erklären Sie die Anzeige!
5. Wie wird das traceroute für die Verbindung von PC01 zu PC08 Ihrer Meinung nach aussehen?



Geben Sie eine begründete Vermutung an! Probieren Sie jetzt traceroute und erklären Sie die Anzeige!







### 4.2.1.3. Clients und Server

Bisher haben wir die Rechner einfach nur verbunden. Jeder Rechner ist gleichwertig. Sie haben nach dem Peer-to-peer-Prinzip (→ [Kommunikations-Konzepte](#), [Peer-to-Peer-Konzept](#)) miteinander kommuniziert. Jeder ist gleichzeitig als Client und Server tätig bzw. kann als solcher fungieren.

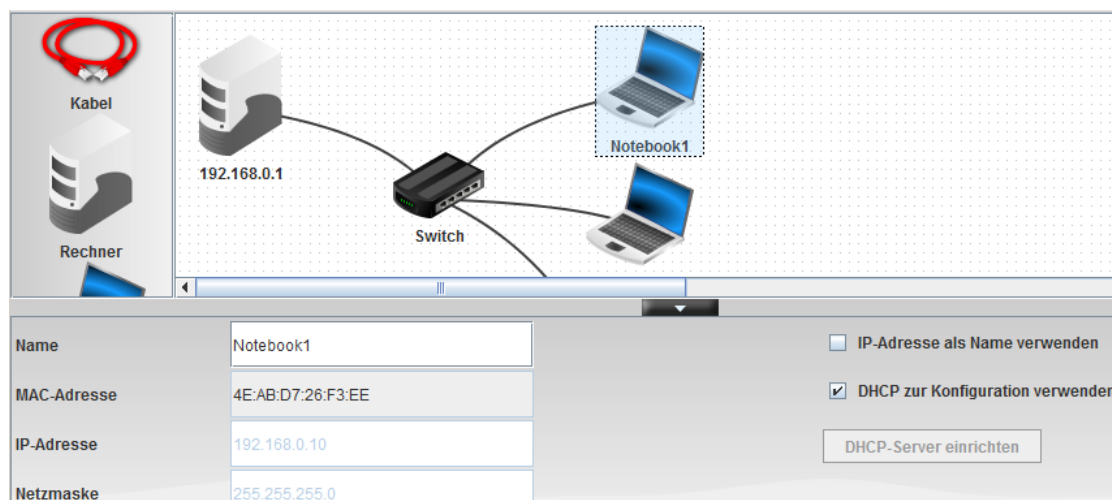
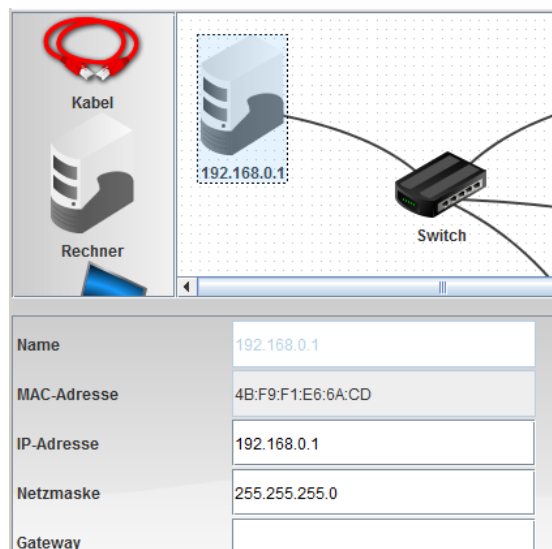
In größeren Netzen brauchen wir aber für bestimmte Aufgaben spezielle Rechner. So sollen z.B. Daten an einem Punkt verwaltet werden. Auch die automatische Bereitstellung von IP-Adressen gehört in diesen Bereich. Natürlich darf nicht jeder einfach irgendwelche Adressen freigeben. Das würde ganz sicher ein Chaos geben. Später kommen auch noch Aufgaben hinzu, die für die Kommunikation mit dem Internet eine Rolle spielen (→ DNS).

Schauen wir uns zuerst die automatische Verteilung von IP-Adresse an. Der Dienst heißt DHCP () und muss auf einem Rechner (Server) laufen. In Filius übernehmen das immer die "großen" Rechner. In der Praxis könnte natürlich auch ein kleiner Laptop als DHCP-Server laufen. Er muss dann nur ständig im Netz verfügbar sein.

Als Beispiel-Netz nehmen wir einen Server (Filius-"Rechner") und einige Client-Notebook's. Dem Server weisen wir eine IP-Adresse zu.

Die angeschlossenen Nutz-Rechner bekommen nur Namen und wir geben bei den Eigenschaften an, dass "DHCP zur Konfiguration verwendet" werden soll.

Die vorher automatisch vergeben IP-Adressen werden nun ausgegraut – sie sind nicht vergeben.



Nachdem alle Rechner einen Namen haben und "wissen", dass sie ihre IP-Adresse über DHCP bekommen sollen, richten wir nun den DHCP-Server ein. Das hätte man natürlich auch beim ersten Einrichten schon machen können, aber jetzt wissen wir auch genau, wieviele IP-Adressen unser Server mindestens bereitstellen muss.

Die Einrichtung des DHCP-Server's ist eigentlich sehr einfach. Im eigenen IP-Netz sucht man sich eine fortlaufende Reihe von IP-Nummern (Unter- und Ober-Grenze) aus, die bisher noch nicht benutzt wird. Ich wähle hier mal absichtlich Nummern, die außerhalb der automatischen Erst-Vorschläge für die Notebook's liegen, damit wir den Effekt auch wirklich sehen können.

Die erste – für DHCP – verfügbare Host-Adresse wäre als die 192.168.0.100 und die letzte 192.68.0.150. Das würde also Platz für 51 Rechner bedeuten.

Die "Statische Adresszuweisung" dient dazu einzelnen Rechnern eine ganz bestimmte Adresse zu reservieren. Häufig brauchen bestimmte Programme eine feste IP, um bestimmte Leistungen zu erbringen. Natürlich kann man solchen Rechnern auch gleich eine feste IP-Adresse zuweisen.

Mit DHCP ist das allerdings etwas eleganter und man kann später schneller einen "Ersatz-Rechner" ins Spiel bringen.

Bei Filius müssen wir hier die MAC-Adressen angeben, bei echten Servern sind das meist die Rechner-Namen.

Das macht dann auch den Austausch eines defekten Rechner (mit statischer DHCP-IP) einfacher, weil der Ersatz einfach den gleichen Namen bekommt, wie der defekte.

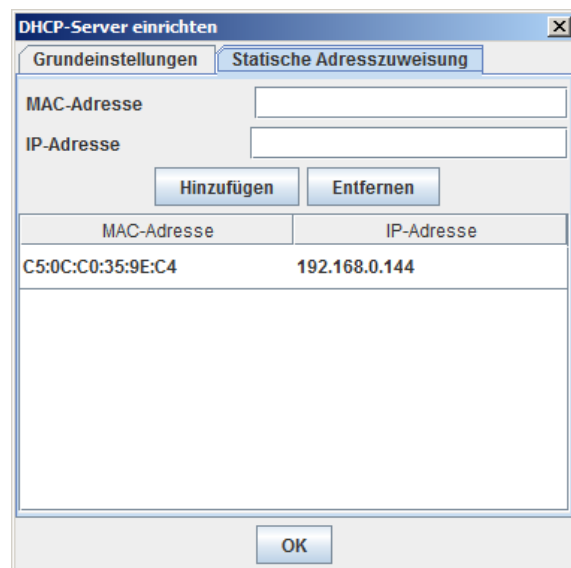
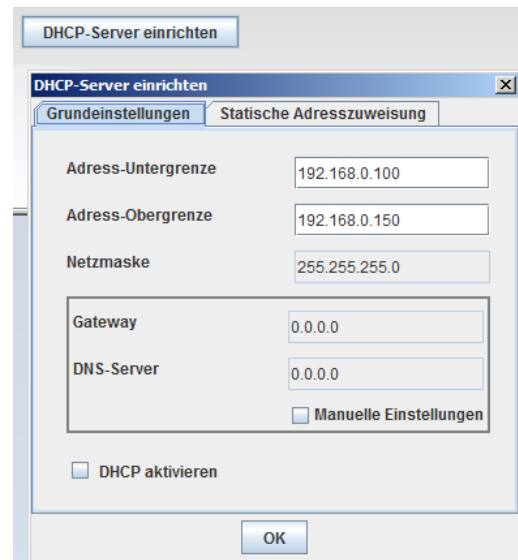
In Filius müssten wir statt dessen eine neue MAC-Adresse eintragen. Eine solche Zuordnung steckt indirekt auch hinter Verwaltung in modernen Servern.

In meinem Beispiel gebe ich dem letzten Rechner (Notebook9) mal eine statische IP innerhalb des DHCP-IP-Bereichs.

Was passiert nun beim Arbeiten im Netz?

Am Ende der Einrichtung müssen wir noch "DHCP aktivieren".

Beim Einschalten (Simulieren) unseres Netzes beobachten wir eine umfangreiche Netzwerk-Aktivität.



Nr.	Zeit	Quelle	Ziel	Protokoll	Schicht	Bemerkungen
1	10:22:49.089	0.0.0.0:68	255.255.255.25...		Anwendung	DHCPDISCOVER yiaddr=0.0.0.0 chaddr=DA:58:F5:39:68:ED
2	10:22:49.095	192.168.0.1:67	255.255.255.25...		Anwendung	DHCPOFFER yiaddr=192.168.0.100 chaddr=DA:58:F5:39:68:ED r...
3	10:22:49.138	0.0.0.0:68	255.255.255.25...		Anwendung	DHCPDISCOVER yiaddr=0.0.0.0 chaddr=4E:AB:D7:26:F3:EE
4	10:22:49.147	192.168.0.1:67	255.255.255.25...		Anwendung	DHCPOFFER yiaddr=192.168.0.101 chaddr=4E:AB:D7:26:F3:EE r...
5	10:22:49.188	0.0.0.0:68	255.255.255.25...		Anwendung	DHCPDISCOVER yiaddr=0.0.0.0 chaddr=C5:0C:C0:35:9E:C4
6	10:22:49.197	192.168.0.1:67	255.255.255.25...		Anwendung	DHCPOFFER yiaddr=192.168.0.144 chaddr=C5:0C:C0:35:9E:C4 r...
7	10:22:49.363	0.0.0.0	192.168.0.101	ARP	Vermittlung	Suche nach MAC für 192.168.0.101, 0.0.0.0: 4E:AB:D7:26:F3:EE
8	10:22:49.413	0.0.0.0	192.168.0.144	ARP	Vermittlung	Suche nach MAC für 192.168.0.144, 0.0.0.0: C5:0C:C0:35:9E:C4
9	10:22:49.463	0.0.0.0	192.168.0.100	ARP	Vermittlung	Suche nach MAC für 192.168.0.100, 0.0.0.0: DA:58:F5:39:68:ED
10	10:22:50.550	0.0.0.0	192.168.0.100	ARP	Vermittlung	Suche nach MAC für 192.168.0.100, 0.0.0.0: DA:58:F5:39:68:ED
11	10:22:50.600	0.0.0.0	192.168.0.101	ARP	Vermittlung	Suche nach MAC für 192.168.0.101, 0.0.0.0: 4E:AB:D7:26:F3:EE
12	10:22:50.663	0.0.0.0	192.168.0.144	ARP	Vermittlung	Suche nach MAC für 192.168.0.144, 0.0.0.0: C5:0C:C0:35:9E:C4
13	10:22:51.801	0.0.0.0:68	255.255.255.25...		Anwendung	DHCPREQUEST yiaddr=0.0.0.0 chaddr=DA:58:F5:39:68:ED reque...
14	10:22:51.801	192.168.0.1:67	255.255.255.25...		Anwendung	DHCPACK yiaddr=192.168.0.100 chaddr=DA:58:F5:39:68:ED ser...
15	10:22:51.852	0.0.0.0:68	255.255.255.25...		Anwendung	DHCPREQUEST yiaddr=0.0.0.0 chaddr=4E:AB:D7:26:F3:EE reque...
16	10:22:51.853	192.168.0.1:67	255.255.255.25...		Anwendung	DHCPACK yiaddr=192.168.0.101 chaddr=4E:AB:D7:26:F3:EE ser...
17	10:22:51.955	0.0.0.0:68	255.255.255.25...		Anwendung	DHCPREQUEST yiaddr=0.0.0.0 chaddr=C5:0C:C0:35:9E:C4 reque...
18	10:22:51.955	192.168.0.1:67	255.255.255.25...		Anwendung	DHCPACK yiaddr=192.168.0.144 chaddr=C5:0C:C0:35:9E:C4 ser...

Jeder DHCP-Client (Notebook1 bis 9) fragt jetzt beim Server nach einer IP-Adresse. Dazu benutzt er einen Broadcast im größten verfügbaren Netz (0.0.0.0/32). Die IP-Port-Nummer für den DHCP-Client-Dienst ist die 68. Der DHCP-Server agiert auf UDP-Port 67.

DHCPDISCOVER

DHCPOFFER

DHCPREQUEST

DHCPACK

## 4.2.2. Verbindung von Netzen

Beginnen wir nun kleine Netze miteinander zu verbinden. Da könnten z.B. in verschiedenen (Schul-)Gebäuden eigene Netze existieren

Die erste spannende Frage beim Betreiben von Netzen könnte sein, ob man verschiedene Netze an einem Switch betreiben kann. Als Beispiel könnte man sich ein normales Arbeits-Netz (z.B. Ausbildungs-Netz der Schule) vorstellen. Daneben soll nun zum Experimentieren ein zweites Netz mit einem eigenen IP-Adress-Bereich aufgebaut werden. So könnte ein Kurs den praktischen Aufbau eines Netzes üben. Leider steht nur ein Switch zur Verfügung. Funktioniert so eine Konstellation, ohne dass die Experimentatoren das Arbeitsnetz stören?

Die beiden Netze bekommen unterschiedliche Klasse C-Adressen und fortlaufende Host-Nummern, die sich auch in den IP-Adressen widerspiegeln.

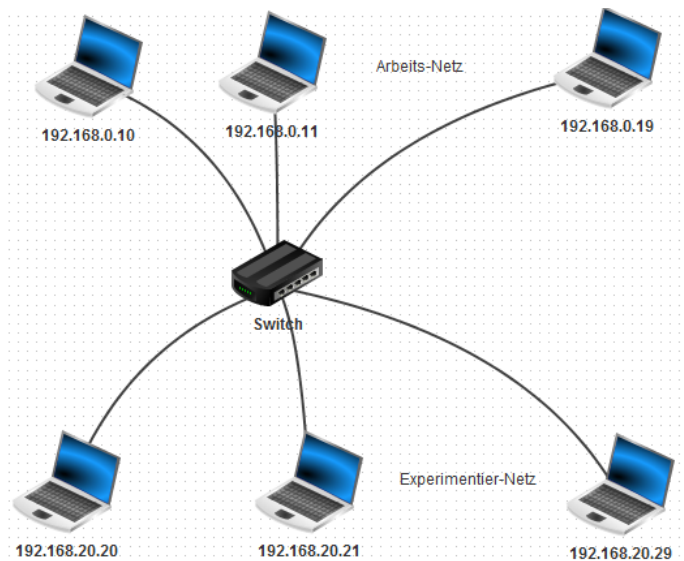
Jedem Netz sind 10 PC's zugeordnet, von denen wir hier jeweils nur 3 darstellen.

Auf den ersten Rechnern der Einzelnetze installieren wir wieder die notwendige Konsole, um die Konnektivität zu testen. Führen wir nun Ping-Versuche innerhalb der Netze durch, dann erhalten wir die üblichen Zeiten. Alles funktioniert, wie erwartet.

Anders sieht das aus, wenn man aus einem der Netze heraus das andere Netz anpingen will.

Obwohl beide Netze über den Switch direkt miteinander verbunden sind, gibt es keine Netz-Verbindungen zwischen ihnen. Der Ping-Befehl bringt ein kurzes "Zieladresse nicht erreichbar" heraus.

Ein Switch eignet sich also scheinbar nicht, um unterschiedliche IP-Netze miteinander zu verbinden. Interessant ist auch, dass der gesamte Datenaustausch leer bleibt.



Zwei Netze an einem Switch

```
root /> ping 192.168.0.19
PING 192.168.0.19 (192.168.0.19)
From 192.168.0.19 (192.168.0.19): icmp_seq=1 ttl=64 time=412ms
From 192.168.0.19 (192.168.0.19): icmp_seq=2 ttl=64 time=200ms
From 192.168.0.19 (192.168.0.19): icmp_seq=3 ttl=64 time=200ms
From 192.168.0.19 (192.168.0.19): icmp_seq=4 ttl=64 time=200ms
--- 192.168.0.19 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust
root /> |
```

Ping im Arbeitsnetz

```
root /> ping 192.168.20.29
PING 192.168.20.29 (192.168.20.29)
From 192.168.20.29 (192.168.20.29): icmp_seq=1 ttl=64 time=402ms
From 192.168.20.29 (192.168.20.29): icmp_seq=2 ttl=64 time=204ms
From 192.168.20.29 (192.168.20.29): icmp_seq=3 ttl=64 time=212ms
From 192.168.20.29 (192.168.20.29): icmp_seq=4 ttl=64 time=200ms
--- 192.168.20.29 Paketstatistik ---
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust
root /> |
```

Ping im Experimentiernetz

```
4 Paket(e) gesendet, 4 Paket(e) empfangen, 0% Paketverlust
root /> ping 192.168.0.10
Zieladresse nicht erreichbar
root />
```

Ping-Versuch vom Experimentiernetz ins Arbeitsnetz

Der Switch wird garnicht angesprochen. Offensichtlich geht die Anfrage gar nicht ins Netz. Die IP-Schicht fängt die Anfrage in ein fremdes Netz schon vorher ab. Anders ist das, wenn man versucht einen unbekanntem (nicht existierenden) Rechner im eigenen anzupingen.

Nr.	Zeit	Quelle	Ziel	Protokoll	Schicht	Bemerkungen
1	16:11:17.148	192.168.20.20	192.168.20.28	ARP	Vermittlung	Suche nach MAC für 192.168.20.28, 192.168.20.20: 35:81:90:...
2	16:11:18.399	192.168.20.20	192.168.20.28	ARP	Vermittlung	Suche nach MAC für 192.168.20.28, 192.168.20.20: 35:81:90:...
3	16:11:19.853	192.168.20.20	192.168.20.28	ARP	Vermittlung	Suche nach MAC für 192.168.20.28, 192.168.20.20: 35:81:90:...
4	16:11:21.104	192.168.20.20	192.168.20.28	ARP	Vermittlung	Suche nach MAC für 192.168.20.28, 192.168.20.20: 35:81:90:...
5	16:11:22.555	192.168.20.20	192.168.20.28	ARP	Vermittlung	Suche nach MAC für 192.168.20.28, 192.168.20.20: 35:81:90:...
6	16:11:23.806	192.168.20.20	192.168.20.28	ARP	Vermittlung	Suche nach MAC für 192.168.20.28, 192.168.20.20: 35:81:90:...
7	16:11:25.258	192.168.20.20	192.168.20.28	ARP	Vermittlung	Suche nach MAC für 192.168.20.28, 192.168.20.20: 35:81:90:...
8	16:11:26.508	192.168.20.20	192.168.20.28	ARP	Vermittlung	Suche nach MAC für 192.168.20.28, 192.168.20.20: 35:81:90:...

Jetzt versucht ARP immerwieder die passende MAC-Adresse zu finden. Da die MAC-Adresse nicht ermittelt werden kann, bekommen wir ein Timeout.

ARP versucht es für jeden Einzel-Ping immer wieder – schließlich könnte ja doch noch eine neue Verbindung entstehen – aber in unserem Fall – mit einem nicht existierenden Rechner – wird das natürlich nichts.

```

root /> ping 192.168.0.10
Zieladresse nicht erreichbar
root /> ping 192.168.20.28
PING 192.168.20.28 (192.168.20.28)
From 192.168.20.28 (192.168.20.28): icmp_seq=1 -- Timeout!
From 192.168.20.28 (192.168.20.28): icmp_seq=2 -- Timeout!
From 192.168.20.28 (192.168.20.28): icmp_seq=3 -- Timeout!
From 192.168.20.28 (192.168.20.28): icmp_seq=4 -- Timeout!
--- 192.168.20.28 Paketstatistik ---
 4 Paket(e) gesendet, 0 Paket(e) empfangen, 100% Paketverlust
root /> |

```

Das der Switch nicht das kritische Teil ist sehen wir auch, wenn wir uns dessen SAT-Tabelle ansehen. Alle angeschlossenen Rechner sind mit ihrer MAC-Adresse dort angemeldet (nach den ersten Netzwerk-Aktivitäten).

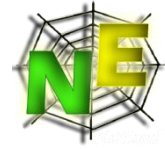
MAC	Port
4E:AB:D7:26:F3:EE	Port 0
C5:0C:C0:35:9E:C4	Port 2
42:D6:52:41:85:F5	Port 5
F5:43:5C:9F:78:B2	Port 4
35:81:90:5B:E0:AF	Port 3
DA:58:F5:39:68:ED	Port 1

Als Lehre können wir nun daraus ziehen, dass wir unterschiedliche Netze ganz beruhigt an einem Switch betreiben können. Die Netze haben keine praktischen Berührungspunkte. Aus Sicherheits-Aspekten heraus sollte man dies aber nicht tun. Ein bössartiger oder unbedarfter Nutzer könnte durch Eingabe (Ausprobieren) anderer – variabler – Netz-Adress-Teile herausbekommen, welche Netze noch vorhanden sind. Mit einer gefundenen Netz-Adresse hätte er dann schon Zugriff auf das andere Netz.

(Also wäre z.B. ein Betreiben des Ausbildungs- und des Verwaltungs-Netzes an einem Switch nicht zulässig, wenn die entsprechende Verordnung dafür eine physikalische Trennung vorschreibt!)

Der Vollständigkeit halber sei gesagt, dass es auch sogenannte "gemanagte" Switches gibt. Bei ihnen lassen sich die Ports am Switch in Gruppen einteilen und die Gruppen wie Einzel-Switches betreiben. Sicher ist das aber auch nicht. Ein einfaches Umstecken eines "bössartigen" Kabels aus einem anderen Netz würde hier alle Sicherheits-Maßnahmen aushebeln.

## 4.3. Simulation von Netzen mit NetEmul



### 4.3.0. Einführung

Download von

verfügbar auf dem IoStick (von HEMPEL)

→ <http://netemul.sourceforge.net/index.html>

### 4.3.1. Programm-Start und Aufbau eines Netzwerkes

Nach dem Start erhält man ein leeres Arbeits-Fenster. Entweder wird nun eine neue Szene ("File" "New") angelegt oder eine vorhandene geöffnet ("File" "Open ..."). Die Szenen basieren auf einem Grund-Raster. Hierhin werden die Geräte positioniert.

Zum Aufbauen oder Erweitern eines Netzes wird einfach das gewünschte Gerät ausgewählt. Nun kann man beliebig viele Geräte dieses Typ's im Gitter ablegen.

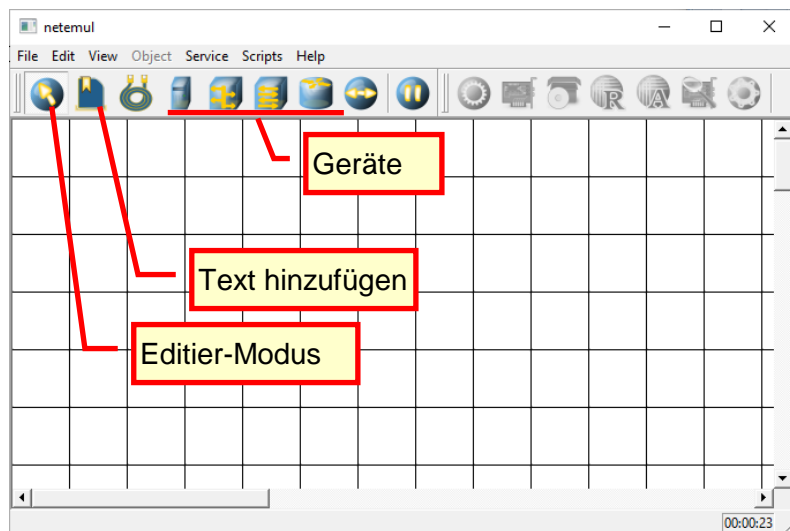
Es werden nur die Grund-Typen von Netz-Geräten unterschieden. Z.B. sind eben Drucker, Smart-TV-Geräte, Laptop's usw. alles nur einfache Netz-Endgeräte, die hier im Programm als "Computer" geführt werden.

Über "Insert text comment" ("Text hinzufügen") lassen sich die Geräte beschriften. Um übersichtliche Netze zu erhalten sollte man von diesem Mittel reichlich Gebrauch machen.

Eine sinnige Beschriftung besteht aus dem Geräte-Namen und z.B. der Netzwerk-Adresse. Dazu gleich mehr.

Der rote Punkt in den Symbolen ist ein Hinweis darauf, dass die Geräte im Netz noch nicht konfiguriert sind.

Wir haben ja auch praktisch die Geräte nur aufgestellt, mit Strom versorgt und ev. gestartet.





Unmittelbar nach dem Ziehen der Verbindung (LAN-Kabel) öffnet sich ein Eigenschaften-Fenster für diese Verbindung. Die gewünschte Vernetzung der beiden Geräte erfolgt über "Connect".

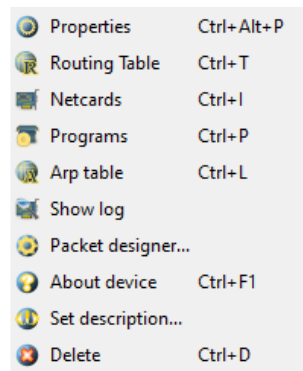
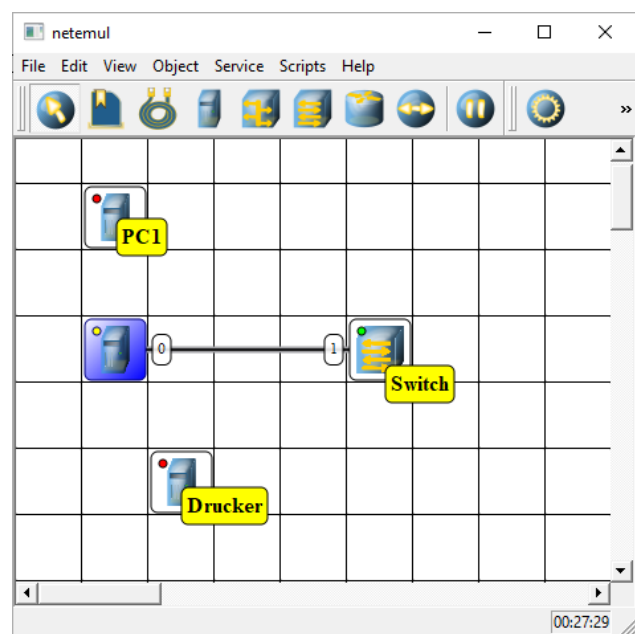
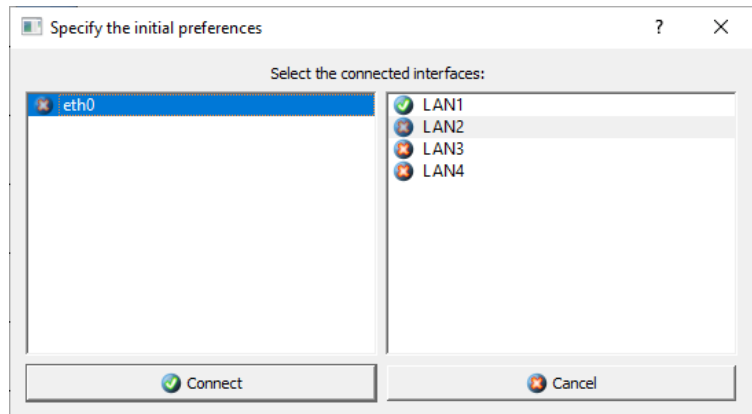
Nun bekommen die Netzgeräte entweder eine gelbe oder grüne Markierung.

Eine grüne Markierung besagt, dass das Gerät schon ausreichend konfiguriert ist.

So muss man z.B. an einem Switch nicht weiter einstellen.

Die Nummern an den Verbindungs-Enden stehen für die benutzte Netzwerkschnittstelle (Netzwerk-Buchse / Netzwerk-Karte).

Gelbe Markierungen weisen darauf hin, dass hier noch Konfigurations-Bedarf besteht. Die Einstellungen erfolgen immer über das Kontext-Menü zu einem Gerät. Dieses erhalten wir – wie üblich – über die rechte Maus-Taste.



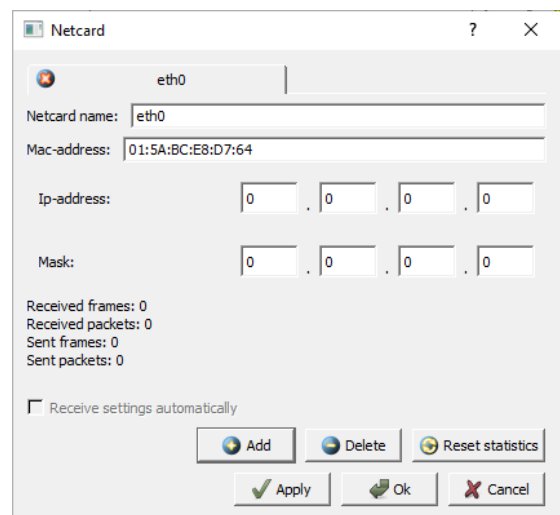
Im Kontext-Menü gehen wir nun auf "Netcards". Der folgende Dialog erwartet jetzt eine sinnvolle IP-Adresse.

Wozu sie dienen und wie die Adressen aufgebaut sind, wird im Abschnitt (→ [8.3.2.1. Internet-Protokoll Version 4 \(IPv4\)](#) ) beschrieben.

Für ein erstes Netz benutzen wir:

Ip-adress: 192.168.0.1  
Mask: 255.255.255.0

Bei weiteren Geräten ändert sich nur die letzte Zahl in der IP-Adresse. Bei der angegebenen Maske können wir hier Zahlen von 1 bis 254 verwenden. Natürlich immer nur einmalig in einem Netz.



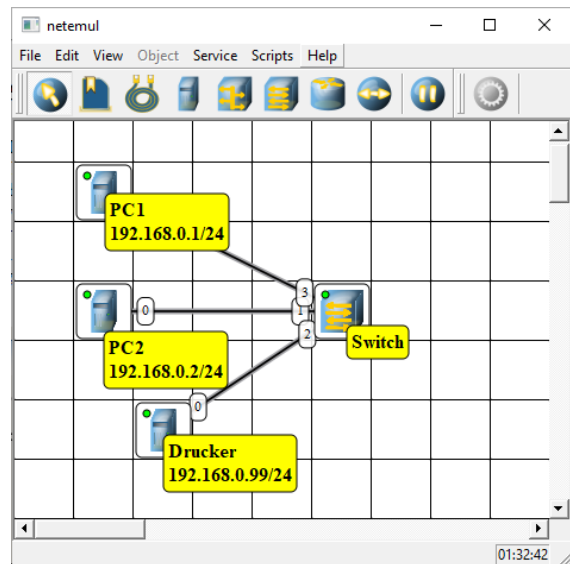
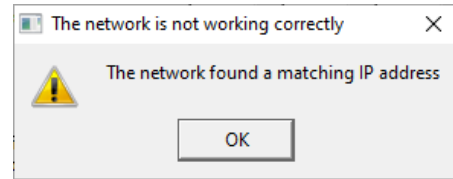
Mit "Apply" wird die Verbindung 'geprüft'. Ist alles in Ordnung, kann mit "Ok" bestätigt werden. Dabei wird ein ARP-Daten-Paket (→ ) an das andere Gerät gesendet. Man erkennt das an einem laufenden Punkt auf der Verbindung. Gelbe Punkte sind Anfragen, grüne Punkte stehen für bestätigende Antworten und rote für Probleme.

Hat man z.B. zwei Geräten die gleiche IP-Adresse zugeteilt, gibt es eine entsprechende Fehler-Meldung. Welche Adresse man dann korrigiert ist egal. Gibt es keine Vorgaben, dann empfehle ich immer eine zum Gerät passende und möglichst fortlaufende Nummerierung.

Auch sollte man die Adresse für Netze mit noch folgenden Simulationen die IP-Adressen mit in die Geräte-Beschreibung übernehmen.

Nach einwenig Übung kann man auf das Prüfen verzichten und sofort mit "Ok" den Dialog verlassen. Die Prüfung erfolgt trotzdem. Dies zeigt uns die ARP-Kommunikation nach einer Erweiterung des Netzes durch eine neue Komponente. Leider werden die Status-anzeigen nicht immer aktualisiert. Es reicht aber, das Netz-Gerät anzuklicken.

Ein kleines lokales (isoliertes) Netzwerk könnte dann z.B. so aussehen (s. Abb. rechts).



### Aufgaben:

- 1. Erstellen Sie mit dem Programm NetEmul das oben beschriebene Netz! Speichern Sie sich dieses als Basis für spätere Simulationen und Erweiterungen extra ab! Arbeiten Sie im Weiteren immer mit einer Kopie dieser Datei!**
- 2. Beobachten Sie die ARP-Kommunikation beim Einrichten oder Konfigurieren einer Netzwerk-Karte (im Konfigurations-Dialog auf "Apply" gehen!)**
- 3. Hätte der Switch nicht auch eine IP-Adresse gebraucht? Erläutern Sie Ihre Meinung dazu!**
- 4. Erweitern Sie das Netz um einen weiteren PC mit der Nummer 7! Verbinden Sie ihn mit dem Switch! Die Netzwerk-Karte soll zuerst einmal eine falsche – schon verwendete – IP-Adresse bekommen. Beobachten Sie den ARP-Datenverkehr in diesem Fall! Notieren Sie Ihre Beobachtungen!**
- 5. Berichtigen Sie die fehlerhafte Adresse und prüfen Sie erneut den ARP-Datenverkehr! Notieren Sie wieder die Beobachtungen!**



### 4.3.2. Simulationen und Beobachtungen im Netzwerk

Mit dem Anklicken einer Netzwerk-Komponente erweitert sich die aktive Funktions-Leiste um diverse Symbole. Diese sind je nach Geräte-Typ unterschiedlich.

Die Eigenschaften ("Show properties") beinhalten lediglich die Möglichkeit einen Gateway einzutragen. Der wird derzeit von uns gar nicht gebraucht.

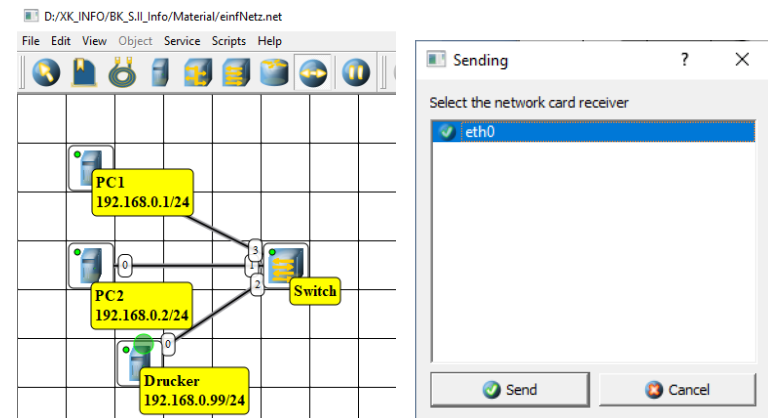
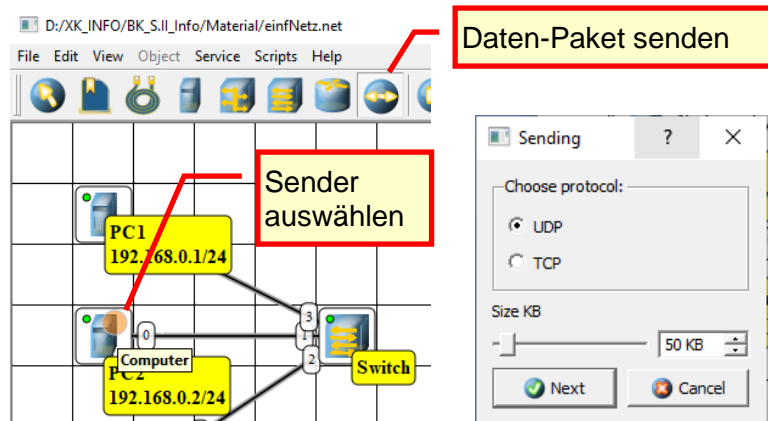
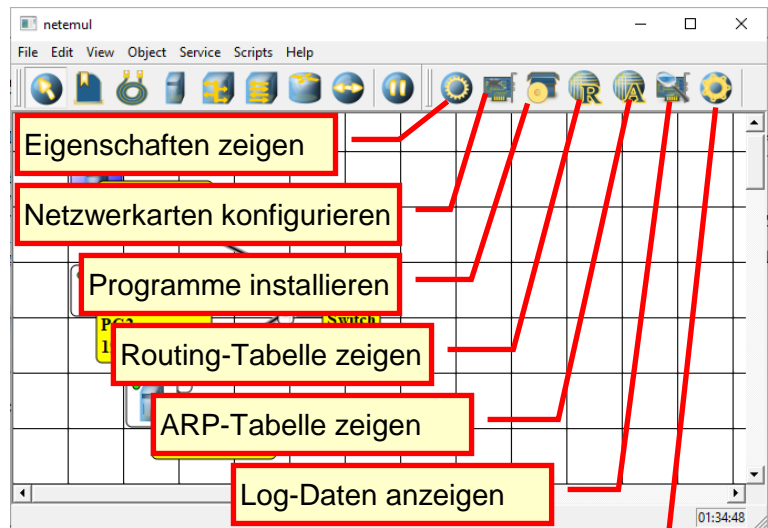
Die Möglichkeit zur Konfiguration der Netzwerk-Karten ("Edit netcards") haben wir vorne schon aufgezeigt.

Im Netz können wir nun auf IP-Ebene Daten verschicken. Da es in NetEmul keine echten Programme auf den Rechnern gibt, übernimmt unsere Simulations-Programme die Erstellung von imaginären Daten.

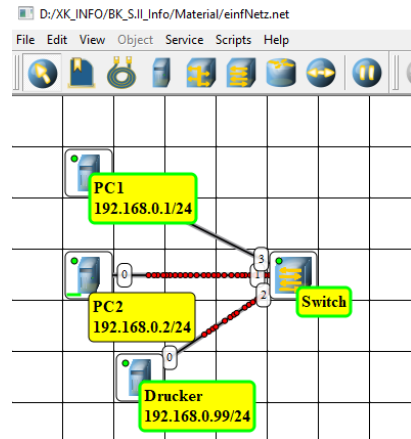
Eine Simulation wird mit der Funktion "Send data" ("Daten-Paket senden") initiiert. Zuerst müssen wir mittels orangemem Cursor den sendenden Computer auswählen. Nach dem Anklicken erscheint ein Dialog, in dem wir die Daten spezifizieren. Dabei geht es um die Festlegung der Datenmenge sowie um die gewünschte Übertragungs-Art.

Danach erscheint ein grüner Cursor, der für die Auswahl des Ziel-Rechner's gedacht ist.

Als letztes muss noch am Ziel-Host die zu verwendende Netzwerk-Karte (-Adresse) ausgeählt werden. Schließlich haben wir die IP-Adresse ja einem einzelnen Netz-Anschluss zugeordnet.



Bevor die eigentlichen Daten in kleineren Paketen (s.a. Abb. rechts: rote Punkte) auf die Reise gehen, wird u.U. zuerst der Ziel-Host abgefragt. Dabei wandert eine Anfrage (gelber Punkt) durch's Netz zu allen Rechnern. Nur der Ziel-Host antwortet mit einer Bestätigung (roter Punkt).



Richtig interessant ist die Verfolgung der Kommunikation anhand der Log-Daten. Diese kann man sich für die verschiedenen Geräte im Netz anzeigen lassen. Es empfiehlt sich hier eine möglichst breite Fenster-Einstellung für NetEmul.

Weiterhin sollte man sich für erste Versuche auf kleine Daten-Mengen beschränken. Die Log-Daten wiederholen sich ab einem besti noch. Da sind kaum neue Erkenntnisse zu gewinnen. Eine recht praktische Daten-Menge sind 5 KByte.

Die Log-Fenster müssen vor der eigentlichen Simulation geöffnet werden ("Show log" über das Kontext-Menü des Gerätes). Damit das Geschehen übersichtlich dargestellt wird, sollte man sich die Log's in der Reihenfolge: Sender, Zwischenstation(en), Empfänger anzeigen lassen.

### Aufgaben:

2. *Arbeiten Sie nun mit einer Kopie dieser Vorlage!*

3.



### kleiner Hilfs-Algorithmus zum Erstellen von Netzwerk's- und Kommunikations-Protokollen

(s.a. nachfolgende Seiten)

1. NetEmul neu starten
2. Netzwerk erstellen
3. Netzwerk speichern
4. NetEmul beenden
5. NetEmul neu starten (damit alle Tabellen, Zeiten, ... zurückgesetzt werden)
6. Netzwerk öffnen
7. Log-Anzeige einstellen ("Show log") für alle geräte, die beobachtet werden sollen
8. Geräte einschalten (z.B. Hinzufügen oder Netzwerk-Karten noch einmal mit "OK" bestätigen, ...)
9. Nachrichten in gewünschter Weise expandieren oder komprimieren
10. Fensterbreite des Haupt-Fenster's sowie der Log's passend machen
11. ev. zusätzliche Info's darstellen (z.B. Switching Tabelle, ...)

Netzwerk-Kommunikation in einem einfachen (lokalen) Netz beim Übertragen von 5 KByte Daten über UDP. Bei einigen Protokollen sind die Zusatz-Informationen ausgeklappt. Die UDP-Nachrichten (grünlich unterlegt) am Ende des Log's wiederholen sich praktisch. Man kann sich auch die Zeit-Stempel der Pakete und Nachrichten ansehen. Da diese aber Realzeit vom Rechner sind, bringt das kaum einen Zusatzwert.

The screenshot shows a network simulation window titled "D:/XK\_INFO/BK\_S\_II\_Info/Material/einfNetz.net". The network diagram features three nodes: PC1 (192.168.0.1/24), PC2 (192.168.0.2/24), and a Drucker (192.168.0.99/24), all connected to a central Switch. Below the diagram are three log windows:

- Computer (PC1):**
  - sent 192.168.0.1 search 192.168.0.99 Type: ARP request
    - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
    - ARP-request:
      - sender IP address: 192.168.0.1
      - sender MAC address: 01:EE:33:AB:D8:3F
      - target IP address: 192.168.0.99
      - target MAC address: 00:00:00:00:00:00
  - received 192.168.0.1 found 192.168.0.99 Type: ARP response
    - Ethernet, sender: 01:5A:BC:E8:D7:64 receiver: 01:EE:33:AB:D8:3F
    - ARP-response:
      - sender IP address: 192.168.0.99
      - sender MAC address: 01:5A:BC:E8:D7:64
      - target IP address: 192.168.0.1
      - target MAC address: 01:EE:33:AB:D8:3F
  - sent 192.168.0.1 >> 192.168.0.99 Type: UDP Message user
    - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: 01:5A:BC:E8:D7:64
    - IP packet, sender: 192.168.0.1, receiver: 192.168.0.99 TTL: 64
    - UDP, sender port: 7777, receiver port: 7777
  - sent 192.168.0.1 >> 192.168.0.99 Type: UDP Message user
  - sent 192.168.0.1 >> 192.168.0.99 Type: UDP Message user
- Switch:**
  - sent 192.168.0.1 search 192.168.0.99 Type: ARP request
    - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
    - ARP-request:
      - sender IP address: 192.168.0.1
      - sender MAC address: 01:EE:33:AB:D8:3F
      - target IP address: 192.168.0.99
      - target MAC address: 00:00:00:00:00:00
- Computer (PC2):**
  - received 192.168.0.1 search 192.168.0.99 Type: ARP request
    - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
    - ARP-request:
      - sender IP address: 192.168.0.1
      - sender MAC address: 01:EE:33:AB:D8:3F
      - target IP address: 192.168.0.99
      - target MAC address: 00:00:00:00:00:00
  - sent 192.168.0.1 found 192.168.0.99 Type: ARP response
    - Ethernet, sender: 01:5A:BC:E8:D7:64 receiver: 01:EE:33:AB:D8:3F
    - ARP-response:
      - sender IP address: 192.168.0.99
      - sender MAC address: 01:5A:BC:E8:D7:64
      - target IP address: 192.168.0.1
      - target MAC address: 01:EE:33:AB:D8:3F
  - received 192.168.0.1 >> 192.168.0.99 Type: UDP Message user
    - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: 01:5A:BC:E8:D7:64
    - IP packet, sender: 192.168.0.1, receiver: 192.168.0.99 TTL: 64
    - UDP, sender port: 7777, receiver port: 7777
  - received 192.168.0.1 >> 192.168.0.99 Type: UDP Message user
  - received 192.168.0.1 >> 192.168.0.99 Type: UDP Message user

Hier die Log-Daten einer TCP-Daten-Übertragung mit einem Umfang von 12 KByte Daten. Geänderte bzw. neuartige Nachrichten sind in explodierter Darstellung angezeigt. Beim ARP-Protokoll habe ich auf eine noch weitere Detailierung verzichtet. Die Informationen gleichen denen bei einer UDP-Daten-Übertragung (s.a. vorige Seite).

The screenshot displays three Wireshark capture windows. The left window, titled 'Computer', shows a sequence of network traffic starting with an ARP request from 192.168.0.1 to 192.168.0.99, followed by an ARP response. This is followed by a series of TCP segments: a sent segment from 192.168.0.1 to 192.168.0.99 and a received segment from 192.168.0.99 to 192.168.0.1. The middle window, titled 'Switch', shows a single ARP request packet from 192.168.0.1 to 192.168.0.99. The right window, titled 'Computer', shows the received ARP response and a series of received TCP segments from 192.168.0.1 to 192.168.0.99. The bottom of each window has a status bar with 'Time', interface name (eth0, LAN1), and a 'Clear' button. The bottom right corner of the rightmost window shows a timestamp of 00:06:22.

### 4.3.2.x. ARP-Nachrichten

In dieser Simulation soll die ARP-Kommunikation beim Zuschalten eines neuen Netzwerk-Gerätes gezeigt werden. Unten im Bildschirm-Ausdruck sind die Log-Daten von PC1, dem Switch und PC2 angezeigt.

Die Netzwerk-Karte an PC1 wurde aktiviert (über ein "OK" bei "Netcards" aus dem Kontextmenü von PC1).

PC1 sendet ein ARP-Paket mit seiner MAC-Adresse an alle Geräte (Broadcast-Adresse FF:..:FF). Im Paket wird die eigene IP-Adresse als Sende- und Empfangs-Adresse mitgeteilt sowie die eigene MAC-Adresse. Die empfangenden Geräte (Switch und PC2 (Drucker natürlich auch)) empfangen das Paket und bauen die Informationen in ihrer ARP-Tabellen ein.

Für den Switch ist die Switching-Tabelle angezeigt. Da der Switch kein IP kann, bleibt bei ihm alles auf dem Niveau der Netzzugriffs-Schicht (bzw. ISO-OSI Schicht 2 (Sicherheit)).

The screenshot shows a network simulation window titled "D:/XK\_INFO/BK\_S.II\_Info/Material/einfNetz.net". The main area displays a network topology with three devices: PC1 (IP 192.168.0.1/24), PC2 (IP 192.168.0.2/24), and a Drucker (Printer), all connected to a central Switch. A "Switching table" window is open, showing a table with the following data:

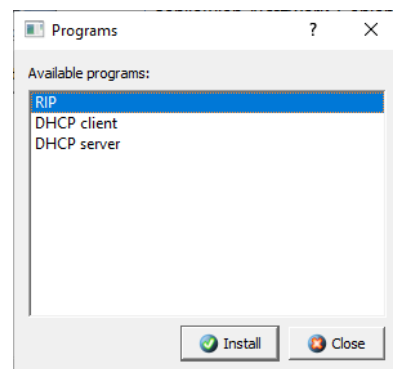
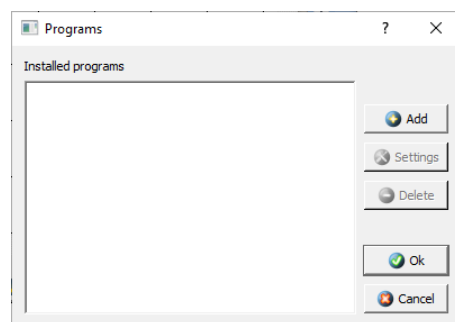
Mac-address	Port	Record type	TTL
01:EE:33:AB:D8:3F	LAN1	Dinamic	170

Below the topology, three log windows show the ARP request details:

- Computer (PC1):** sent 192.168.0.1 search 192.168.0.1 Type: ARP request. Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF. ARP-request: sender IP address: 192.168.0.1, sender MAC address: 01:EE:33:AB:D8:3F, target IP address: 192.168.0.1, target MAC address: 00:00:00:00:00:00.
- Switch:** received 192.168.0.1 search 192.168.0.1 Type: ARP request. Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF. ARP-request: sender IP address: 192.168.0.1, sender MAC address: 01:EE:33:AB:D8:3F, target IP address: 192.168.0.1, target MAC address: 00:00:00:00:00:00.
- Computer (PC2):** received 192.168.0.1 search 192.168.0.1 Type: ARP request. Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF. ARP-request: sender IP address: 192.168.0.1, sender MAC address: 01:EE:33:AB:D8:3F, target IP address: 192.168.0.1, target MAC address: 00:00:00:00:00:00.

### 4.3.2.x. IP-Nachrichten

### 4.3.2.x. zusätzliche Programme auf den Netzgeräten



Netze können auch per Skript definiert ("aufgebaut") werden.

Nebenstehendes Skript ergibt das unten abgebildete Netzwerk.

Das Skript ist eine Version eines originalen Skriptes ("arp.js"), die zum Programm mitgeliefert wird. Aus dieser Datei wurden die russischen Kommentare entfernt.

Wenn man einige Grundkenntnisse in der Programmierung und vielleicht auch in JavaScript hat, dann kann man die einzelnen Abschnitte nachvollziehen.

Für unsere einfachen Zwecke ist eine Notierung von Netzen in Skript-Form aber nicht notwendig.

```
arp_y.js - Editor
Datei Bearbeiten Format Ansicht Hilfe
myCanvas::emulateTime();
emulateTime();

var router = addRouter(5,5);
router.setIp("LAN10", "192.168.1.2");
if ( open ) closeScene();
newScene();
stop();

var sw = new Array();
sw[0] = addSwitch(3,3);
sw[0].setSocketsCount(8);
sw[1] = addSwitch(7,3);
sw[1].setSocketsCount(8);

var r = addRouter(5,3);
r.router = true;
r.setSocketsCount(8);
addConnection(sw[0],r,"LAN8","LAN3");
addConnection(sw[1],r,"LAN8","LAN4");

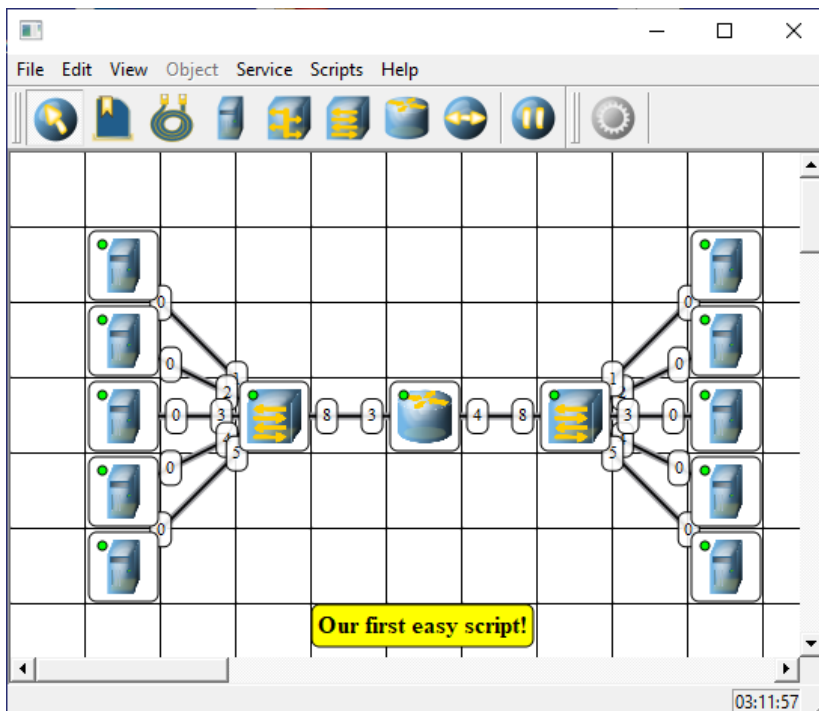
var net = new Array();
var x = 1 , g = 126;
for ( j = 0 ; j < 2 ; j++ ) {
  net[j] = new Array();
  for ( i = 1 ; i <= 5 ; i++ ) {
    net[j][i-1] = addComputer(x,i);
    addConnection( net[j][i-1] , sw[j] , "eth0" , "LAN"+i );
    net[j][i-1].setIp("eth0", "192.168.1."+( i + j*128) );
    net[j][i-1].setMask("eth0", "255.255.255.128");
    net[j][i-1].setGateway("192.168.1."+g);
  }
  r.setMask("LAN"+(j+3) , "255.255.255.128");
  r.setIp("LAN"+(j+3) , "192.168.1." + g );
  x += 8;
  g += 128;
}

net[0][0].sendMessage("192.168.1.133",50,0);
emulateTime();
play();

result = ( net[0][0].sendPacketCount("eth0") == net[1][4].receivePacketCount("eth0") );

var text = addNote(4,6);
text.note = "Our first easy script!";
saveScene("test/arp.net");

result;
```





### 4.3.3. Erweiterung um weitere Netzwerk-Komponenten und -Verbindungen

#### 4.3.3.1. Einbau eines Server's zum Bereitstellen eines Dienstes

Die meisten Heim-Netzwerke und deren familiären Installateure haben meist noch nie was von IP-Adressen gehört. Sie werden nie irgendwo festgelegt und wenn man mit einem Computer in ein anderes Netz (z.B. auf Arbeit oder in der Schule) umzieht, dann funktioniert das Netz einschließ IP ganz wie von Zauberhand. Aber woher bekommen die Rechner dann immer ihre IP-Adressen?

Verantwortlich dafür ist ein Dienst der DHCP heißt. Genauer besprechen wir diesen im Abschnitt → .

Im Netz muss ein Gerät vorhanden sein, das die IP-Adressen verteilt. Dies ist der sogenannte DHCP-Server. Die anderen geräte können dann DHCP-Client's sein und vom Server beim Verbindungs-Aufbau eine IP-Adresse zugeteilt bekommen.

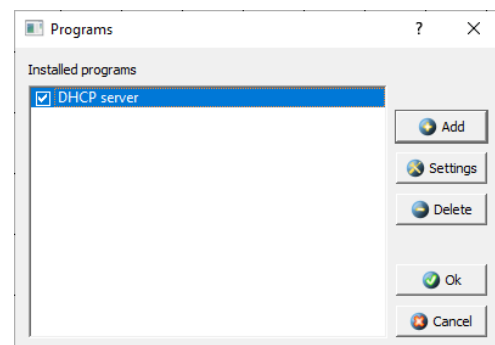
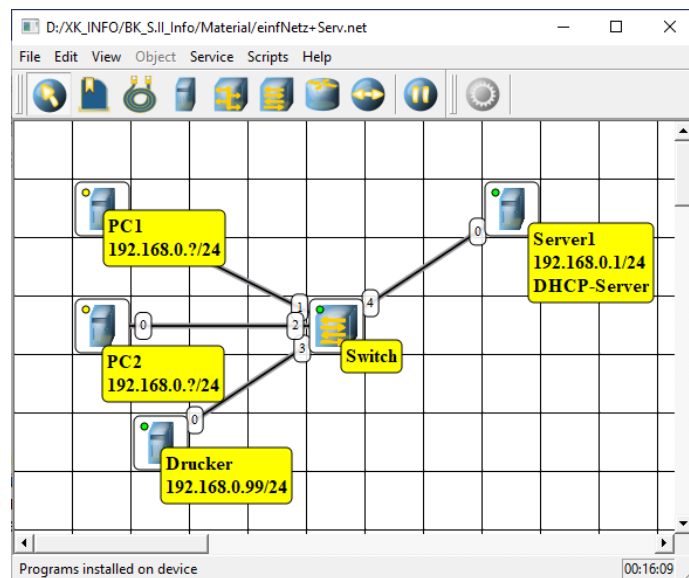
Meist ist der DHCP-Server gleich im Router angelegt. Dieser läuft ja schließlich 24 Stunden durch. In professionellen Netzen mit einem Hardware-Server ist die Software des DHCP-Server's hier installiert. Praktisch jedes Server-Betriebssystem bringt einen solchen Server mit.

Zuerst wollen wir mal so einen Server in unser Netz bringen.

Sachlich ist er ein normalers Netzwerk-Endgerät. Wir nennen ihn hier Server1. Er muss zwingend eine IP-Adresse bekommen. Zum Server wird er dadurch, dass wir nun einen Service auf dem "Server" installieren. Das haben wir prinzipiell schon weiter vorne beschrieben.

Hier muss nun eine Einrichtung der Adress-Verteilung vorgenommen werden.

Dazu aktivieren wir in der Programm-Liste den Service und gehen dann auf "Settings" ("Einstellungen")





Der interessante Teil ist unten der "Dynamic"-Bereich. Er muss meist auch aktiviert werden, sonst läuft der Service, verteilt aber keine IP-Adressen.

Wir legen hier den Adress-Bereich (von / bis) fest. Ich habe hier mal die Host-Adressen 100 bis 199 dafür eingeplant.

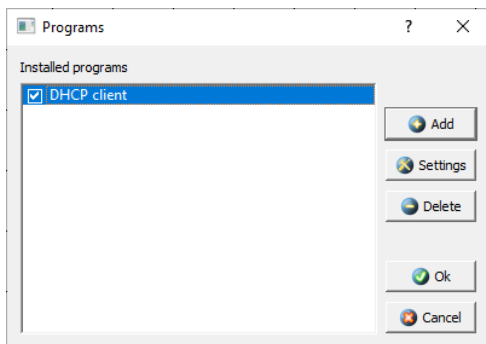
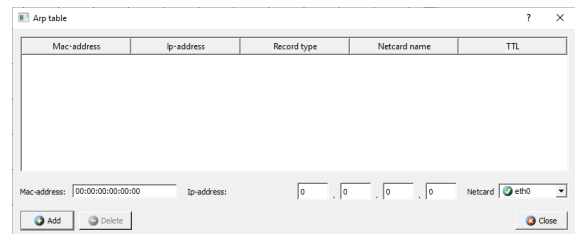
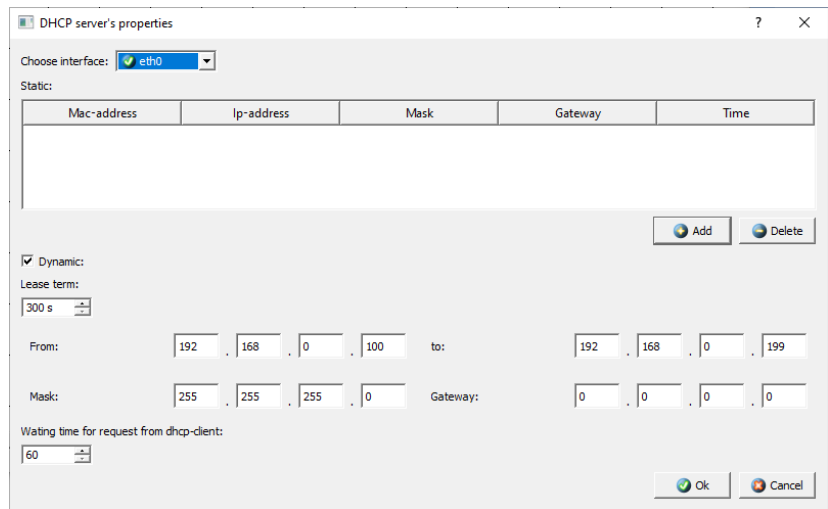
DHCP-Client's bekommen nun aus diesem Adress-Bereich eine Adresse, wenn sie diese anfordern.

Im oberen Fenster-Bereich kann man feste IP-Adressen vergeben. Immer wenn das Gerät mit der entsprechenden MAC-Adresse eine IP anfordert, bekommt es genau diese.

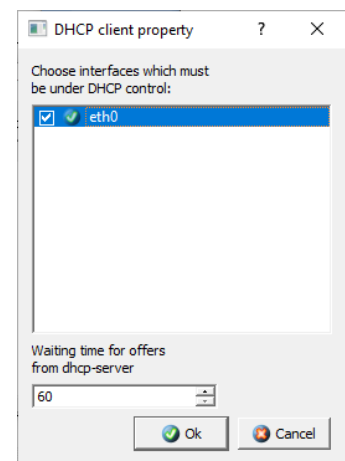
Trotz alledem kann man die anderen IP-Adressen auch immer noch statisch (am Gerät) selbst festlegen. Überschneidungen der Adressen müssen aber unbedingt vermieden werden, ansonsten kann zu IP-Fehlern kommen.

Bei der Anzeige der ARP-Tabelle am Server ist diese leer. Der Server kennt noch keinen weiteren Rechner.

An den IP-Adress-losen Rechnern müssen wir nun den DHCP-Client installieren.



Die Einstellungen am Client sind deutlich einfacher. Es muss nur die zu verwendene Netzwerk-Schnittstelle ausgewählt werden.



Was jetzt im Netz abgeht, ist schon beachtlich. Im Log-Bereich ist links der Client (PC1) und rechts der Server (Server1) dargestellt.

Zu beachten ist, dass man die Nachrichten immer abwechselnd bei den beiden Geräten lesen muss, um den Daten-Austausch nachzuverfolgen. Aber das kennen wir ja schon.

Am Besten ist es auch die Situation nachzustellen und den Nachrichten-Austausch schrittweise zu beobachten und zu deuten.

The screenshot displays a network simulation environment. At the top, a window titled "D:/XK\_INFO/BK\_S\_II\_Info/Material/einfNetz+Serv.net" shows a network diagram with four nodes: PC1 (IP 192.168.0.2/24), PC2 (IP 192.168.0.2/24), a central Switch, and Server1 (IP 192.168.0.1/24 DHCP-Server). Below the diagram are two "Computer" windows showing network traffic logs.

**Left Computer Window (PC1):**

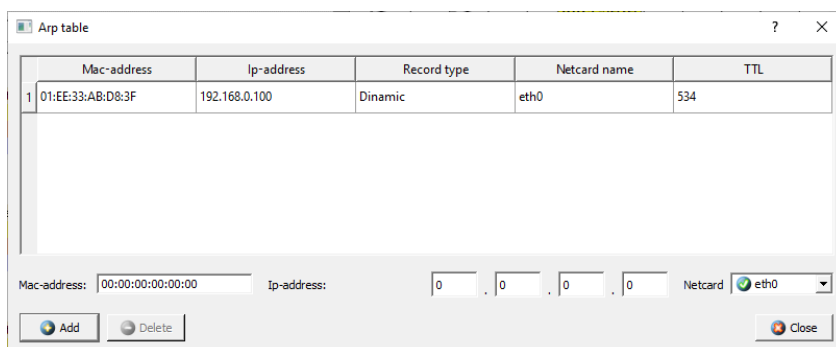
- sent 0.0.0.0 >> 255.255.255.255 Type: DHCP message
  - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
  - IP packet, sender: 0.0.0.0, receiver: 255.255.255.255 TTL: 64
  - UDP, sender port: 67, receiver port: 68
  - DHCP Message, Type: DHCPDISCOVER
    - Xid: 1245, Yiaddr: 0.0.0.0
    - Siaddr: 0.0.0.0, Chaddr: 01:EE:33:AB:D8:3F
- received 192.168.0.1 >> 255.255.255.255 Type: DHCP message
  - Ethernet, sender: 01:B2:D8:46:50:62 receiver: FF:FF:FF:FF:FF:FF
  - IP packet, sender: 192.168.0.1, receiver: 255.255.255.255 TTL: 64
  - UDP, sender port: 67, receiver port: 68
  - DHCP Message, Type: DHCPOFFER
    - Xid: 1245, Yiaddr: 192.168.0.100
    - Siaddr: 192.168.0.1, Chaddr: 01:EE:33:AB:D8:3F
- sent 0.0.0.0 >> 255.255.255.255 Type: DHCP message
  - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
  - IP packet, sender: 0.0.0.0, receiver: 255.255.255.255 TTL: 64
  - UDP, sender port: 67, receiver port: 68
  - DHCP Message, Type: DHCPREQUEST
    - Xid: 1245, Yiaddr: 0.0.0.0
    - Siaddr: 192.168.0.1, Chaddr: 01:EE:33:AB:D8:3F
- received 192.168.0.1 >> 255.255.255.255 Type: DHCP message
  - Ethernet, sender: 01:B2:D8:46:50:62 receiver: FF:FF:FF:FF:FF:FF
  - IP packet, sender: 192.168.0.1, receiver: 255.255.255.255 TTL: 64
  - UDP, sender port: 68, receiver port: 67
  - DHCP Message, Type: DHCPACK
    - Xid: 1245, Yiaddr: 192.168.0.100
    - Siaddr: 192.168.0.1, Chaddr: 01:EE:33:AB:D8:3F
- sent 192.168.0.100 search 192.168.0.100 Type: ARP request
  - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
  - ARP-request:
    - sender IP address: 192.168.0.100
    - sender MAC address: 01:EE:33:AB:D8:3F
    - target IP address: 192.168.0.100
    - target MAC address: 00:00:00:00:00:00

**Right Computer Window (Server1):**

- received 0.0.0.0 >> 255.255.255.255 Type: DHCP message
  - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
  - IP packet, sender: 0.0.0.0, receiver: 255.255.255.255 TTL: 64
  - UDP, sender port: 67, receiver port: 68
  - DHCP Message, Type: DHCPDISCOVER
    - Xid: 1245, Yiaddr: 0.0.0.0
    - Siaddr: 0.0.0.0, Chaddr: 01:EE:33:AB:D8:3F
- sent 192.168.0.1 >> 255.255.255.255 Type: DHCP message
  - Ethernet, sender: 01:B2:D8:46:50:62 receiver: FF:FF:FF:FF:FF:FF
  - IP packet, sender: 192.168.0.1, receiver: 255.255.255.255 TTL: 64
  - UDP, sender port: 68, receiver port: 67
  - DHCP Message, Type: DHCPOFFER
    - Xid: 1245, Yiaddr: 192.168.0.100
    - Siaddr: 192.168.0.1, Chaddr: 01:EE:33:AB:D8:3F
- received 0.0.0.0 >> 255.255.255.255 Type: DHCP message
  - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
  - IP packet, sender: 0.0.0.0, receiver: 255.255.255.255 TTL: 64
  - UDP, sender port: 67, receiver port: 68
  - DHCP Message, Type: DHCPREQUEST
    - Xid: 1245, Yiaddr: 0.0.0.0
    - Siaddr: 192.168.0.1, Chaddr: 01:EE:33:AB:D8:3F
- sent 192.168.0.1 >> 255.255.255.255 Type: DHCP message
  - Ethernet, sender: 01:B2:D8:46:50:62 receiver: FF:FF:FF:FF:FF:FF
  - IP packet, sender: 192.168.0.1, receiver: 255.255.255.255 TTL: 64
  - UDP, sender port: 68, receiver port: 67
  - DHCP Message, Type: DHCPACK
    - Xid: 1245, Yiaddr: 192.168.0.100
    - Siaddr: 192.168.0.1, Chaddr: 01:EE:33:AB:D8:3F
- received 192.168.0.100 search 192.168.0.100 Type: ARP request
  - Ethernet, sender: 01:EE:33:AB:D8:3F receiver: FF:FF:FF:FF:FF:FF
  - ARP-request:
    - sender IP address: 192.168.0.100
    - sender MAC address: 01:EE:33:AB:D8:3F
    - target IP address: 192.168.0.100
    - target MAC address: 00:00:00:00:00:00

Am Ende hat der PC1 eine IP-Adresse erhalten. In diesem Fall die Host-Adresse 100. Mit diesem kann er nun wieder einen ARP-Datenaustausch durchführen. Die ordentliche Intergration ins IP-Netz sehen wir am grünen Status-Symbol am PC1 (ev. einmal auf die Simulations-Fläche zum Aktualisieren klicken).

Die ARP-Tabelle des Servers verzeichnet nun auch das neue Netzgerät.



	Mac-address	Ip-address	Record type	Netcard name	TTL
1	01:EE:33:AB:D8:3F	192.168.0.100	Dinamic	eth0	534

Mac-address: 00:00:00:00:00:00 Ip-address: 0 . 0 . 0 . 0 Netcard: eth0

Add Delete Close

### **Aufgaben:**

- 1. Erstellen ein Netzwerk aus 3 PC's und einem Drucker sowie einem Server (Server1)!***
- 2. Installieren Sie auf dem Server einen DHCP-Dienst nach eigenem Ermessen!***
- 3. Aktivieren Sie die Log-Funktion für den Server und mindestens einem Client!***
- 4. Versehen Sie nun die Endgeräte mit einer statischen IP-Adresse oder einem DHCP-Client!***
- 5. Übernehmen Sie die Log-Daten in Ihre Aufzeichnungen!***
- 6. Erläutern Sie Schritt-weise den Datenaustausch!***

### **4.3.3.2. Einbau eines Router's zum Verbindung von unterschiedlichen Netzen**

### **4.3.3.3. Einbau einer Bridge zum Verbindung von Teil-Netzwerk**

---

Die Ethernet-Technologie legt bestimmte Parameter bezüglich der maximalen Kabel-Längen vor. Was aber, wenn man eine große Firma vernetzen will und trotzdem alle Computer miteinander kommunizieren können sollen.

Das Gerät für die Verbindung von Teil-Netzwerken ist eine sogenannte Bridge (→ ). Aber eine Bridge suchen wir in NetEmul vergeblich.

Wenn Sie sich schon intensiver mit der Theorie beschäftigt haben, dann wissen Sie, dass eine Bridge nur ein spezieller Router ist. Ganzgenau eigentlich eine Kombination aus zwei Routern, die über die externe Anschlussstelle direkt miteinander verknüpft sind.

#### **4.3.3.4. Verwendung eines Gateway's – zentrale Kontrolle**

#### **4.3.3.5. Verwendung eines Hub's in Netzen**

Hub's werden in modernen Netzen kaum noch verwendet. Da sie immer nur eine Netzwerk-Verbindung bedienen können, sind sie nicht mehr leistungs-fähig genug. Die Leistungs-fähigere Weiterentwicklung ist der Switch, den wir schon benutzt haben. Erstellt die aktuelle Technik dar.

Für uns ist aber die Beobachtung der Arbeitsweise interessant.

alle Rechner eines Netzes an einem Hub bilden eine Kollisions-Domäne  
Stern-Topologie bewirkt so, das nur eine einzige Verbindung im Stern möglich ist

Switche können ein Netz in mehrere Kollisions-Domänen aufteilen  
dadurch sind mehr quasi parallele Verbindungen innerhalb des Sterns möglich

---

***(komplexe) Aufgaben (zur Vorbereitung auf eine Klausur od.ä.):  
die blauen Aufgaben sind ergänzend für die gehobene Anspruchsebene***

***1. Erstellen Sie mit dem Programm NetEmul zwei zuerst einmal getrennte Netze!***

***Beobachten Sie nach dem Hinzufügen des jeweils letzten Gerät's in den beiden Netzen die ARP-Kommunikation! Lassen Sie sich dazu gleich nach dem Positionieren des Gerätes die Log's vom Gerät und vom Netzkoppler anzeigen!***

***Netz1 (linke Seite auf Arbeitsfläche) besteht aus drei PC's (PC1, ...), die an einem Hub angeschlossen sind. Die Rechner bekommen fortlaufende Host-Adressen (1, ...) in einem 192.168.10.0/24-Netz.***

***Das Netz2 besteht aus drei Servern (Serv1, ...), welche die Adressen 10.10.0.32, 10.2.20.14 und 10.1.1.1 in einem /8-Netz haben. Sie sind über ein Switch miteinander verbunden. Zwischen beiden Netzen stellen Sie ein Hub, ein Switch und ein Router bereit, ohne diese mit den beiden seitlichen Netzen zu verbinden. Speichern Sie sich dieses Modell als Vorlage ab!***

***2.0. Aktivieren Sie die Darstellung der Log's für PC1, den Hub und PC3!***

***2.1. Versenden Sie vom PC1 ein 12 KByte-Datenpaket per UDP an PC3!***

***2.2. Beschreiben Sie den Daten-Austausch (wandernde Pakete)!***

***2.3. Wiederholen Sie das Versenden mehrfach (mind. 2x) und beobachten sowie beschreiben Sie Ihre Beobachtungen!***

***2.4. Erläutern Sie die ausgetauschten Daten lt. Log's!***

***3.0. Aktivieren Sie die Darstellung der Log's für PC2, den Hub und PC1!***

***3.1. Versenden Sie im vom PC2 ein 12 KByte-Datenpaket per TCP an PC1!***

***3.2. Beschreiben Sie den Daten-Austausch!***

***3.3. Wiederholen Sie das Versenden mehrfach (mind. 2x) und beobachten sowie beschreiben Sie Ihre Beobachtungen!***

***3.4. Erläutern Sie die ausgetauschten Daten lt. Log's!***

***4. Erläutern Sie die Funktionsweise eines Hub's! Gehen Sie auf die benutzten Schichten des ISO-OSI-Modell's (und des TCP/IP-Modell's) ein! Erstellen Sie auch eine Türmchen-Skizze (Nummerierung der Schichten mit 1 bis 7 reicht aus)!***

***5.0. Aktivieren Sie die Darstellung der Log's für Serv1, den Switch und Serv3!***

***5.1. Versenden Sie vom Serv1 ein 12 KByte-Datenpaket per UDP an Serv3!***

***5.2. Beschreiben Sie den Daten-Austausch!***

***5.3. Wiederholen Sie das Versenden mehrfach (mind. 2x) und beobachten sowie beschreiben Sie Ihre Beobachtungen!***

***5.3. Erläutern Sie die ausgetauschten Daten lt. Log's!***

***6.0. Aktivieren Sie die Darstellung der Log's für Serv2, den Switch und Serv1!***

***6.1. Versenden Sie vom Serv2 ein 12 KByte-Datenpaket per TCP an Serv1!***

***6.2. Beschreiben Sie den Daten-Austausch!***

***6.3. Wiederholen Sie das Versenden mehrfach (mind. 2x) und beobachten sowie beschreiben Sie Ihre Beobachtungen!***

***6.4. Erläutern Sie die ausgetauschten Daten lt. Log's!***

- 
7. Erläutern Sie die Funktionsweise eines Switch's! Gehen Sie auf die benutzten Schichten des ISO-OSI-Modell's (und des TCP/IP-Modell's) ein! Erstellen Sie auch eine Türmchen-Skizze!
  8. Verbinden Sie in der Modell-Vorlage die beiden Netze über die zur Verfügung stehenden Netzkoppel-Geräte! Richten Sie die Koppel-Geräte nach den jeweiligen Möglichkeiten ein, so dass, wenn es geht, beide Netze verbunden sind! Speichern Sie jedes Modell in einer extra Datei mit entsprechendem Namen!
  - 9.0. Aktivieren Sie die Darstellung der Log's für PC3, das Netze-verbindende Koppel-Gerät und Serv1!
  - 9.1. Versenden Sie vom PC3 ein 12 KByte-Datenpaket per TCP an Serv1!
  - 9.2. Beschreiben Sie den Daten-Austausch (wandernde Pakete)!
  - 9.3. Wiederholen Sie das Versenden mehrfach (mind. 2x) und beobachten sowie beschreiben Sie Ihre Beobachtungen!
  - 9.4. Erläutern Sie die ausgetauschten Daten lt. Log's!
  - 9.5. Versenden Sie vom Serv1 ein 12 KByte-Datenpaket per TCP an PC3!
  - 9.6. Beschreiben Sie den Daten-Austausch (wandernde Pakete)!
  10. Erläutern Sie die Funktionsweise eines Router's! Gehen Sie auf die benutzten Schichten des ISO-OSI-Modell's (des DoD- und des TCP/IP-Modell's) ein! Erstellen Sie auch eine Türmchen-Skizze!

---

## 4.x. spezielle Netze und Protokolle

### ISDN

→ KALDEALI → S. 126 ff.

### DSL

#### **DSL-Techniken**

- **ADSL**  
asymmetric DSL bis 8 Mbit/s Download + 1 Mbit/s Upload
- **HDSL**  
high data rate DSL bis 2 Mbit/s Download
- **SDSL**  
symmetric DSL bis 3 Mbit/s Down- und Upload
- **VDSL**  
very high data rate DSL bis 52 Mbit/s Download + 2,3 Mbit/s Upload

neue Anschlüsse praktisch nur noch VoIP-Telefonie

Nachteil es entfällt die unabhängige Stromversorgung der Telefonsysteme aus sich selbst heraus, bei Strom-Ausfall liegt gesamte Kommunikations-Struktur am Boden  
alte analoge Telefon-Leitungen hatten auch Strom, wenn Netz ausfällt → gut auch für Notfall-Telefone bei Schwerkranken etc.

### Kabelmodem

Internet-Versorgung über Fernseh-Kabel-Anschluss

oft kombiniert mit VoIP-Telefonie

---

## WLAN

→ KALDEALI → S. 287 ff.

## GSM

→ KALDEALI → S. 241 ff.

global standard for mobile communication  
seit 1985 in Deutschland

2. Generation der Handy-Telefonie

1. Generation waren analoge Funknetze, abgelöst durch digitale Funknetze (2. Generation)  
3. Generation ist Breitband-Technik

GSM

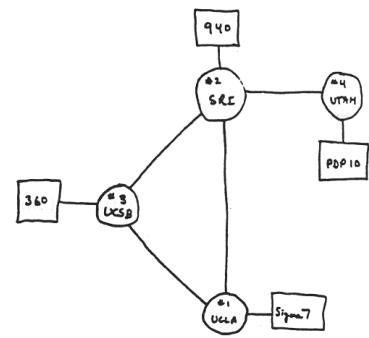
900 MHz (D1 und D2) oder 1800 MHz (E+ und E2)  
rund 10 kbit/s relativ langsam → GPRS-Technologie (general packet radio service)

## UMTS

3. Generation der Handy-Netze  
universal mobile telecommunication system



# Internet



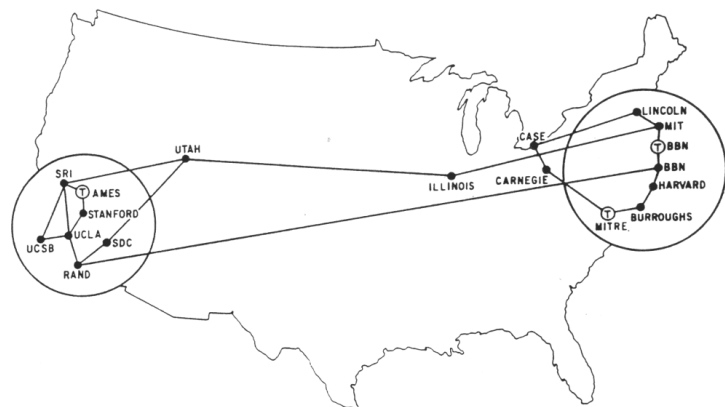
THE ARPA NETWORK

DEC 1969

4 NODES

FIGURE 6.2 Drawing of 4 Node Network  
(Courtesy of Alex McKenzie)

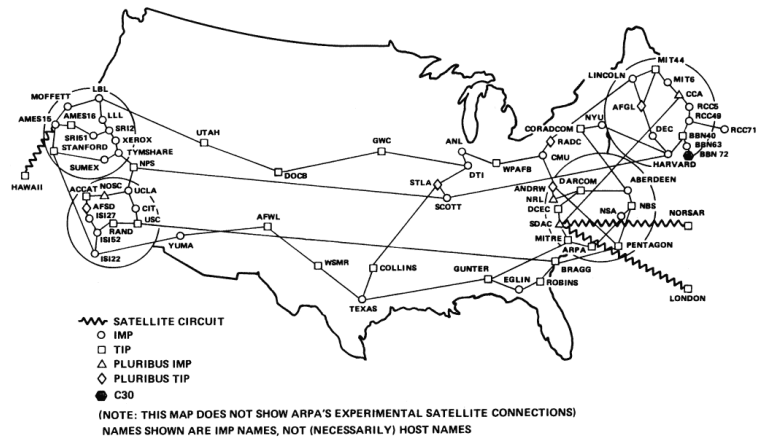
Ideen-Skizze zum ARPA-Netz



MAP 4 September 1971

ARPA-Netz im Jahre 1971

ARPANET GEOGRAPHIC MAP, OCTOBER 1980



ARPA-Netz im Jahre 1980



Q: ??? (unbekannt)

→ KALDEALI → S. 308 ff.

---

## Biographie: Tim BERNERS-LEE (1955 - )

HTML  
www

---

## TCP/IP

→ KALDEALI → S. 310 ff.

→ KALDEALI → S. 317 ff.

## IPv4

## IPv6

Einteilung der IPv6-Adressen

→ KALDEALI → S. 341 ff.

```
Network Identifier Interface Identifier
                        Host Identifier
2001:0db8:0000:0000:03a4:0000:0000:9f21
2001:db8:0:0::/64
```

der Documentation Prefix 2001:db8::/64 muss im konkreten Netz durch die (vom RIPE zugewiesene) Netzwerk-Adresse ersetzt werden

den blauen Adressteil kann jeder Besitzer eines RIPE-Netzwerkes frei gestalten dort stehen jetzt 4,3 Mrd. Netzwerke zur Verfügung, also praktisch so viele, wie heute theoretisch mit IPv4 insgesamt möglich wären

für jeden Standort sollte ein /48-Netzwerk zugeordnet werden

kleinere Netzwerke z.B. ein /56-Netzwerk für 5 Clients sind möglich. aber nicht empfohlen

Empfehlungen für die Aufteilung des Standort-Netzwerkes (Best practice):

- für die Infrastruktur 4x /56-Netzwerke → 1024 Netzwerke
  - z.B. für Router, Syslog-Server, ...
- freigehaltener Bereich 12x /56 → 3072 Netzwerke
  - als Reserve
- Bereich für DMZ 4x /56-Netzwerke → 1024 Netzwerke
  - z.B. für 1024 separate DMZ-Rechner
- freigehaltener Bereich 12x /56 → 3072 Netzwerke

- als Reserve

Zwischenstand: 8192 Netzwerke verbraucht / verplant / ...

für Gebäude 1:

- für die Infrastruktur 4x /56-Netzwerke → 1024 Netzwerke
  - z.B. für Clients, Drucker, ...
- freigehaltener Bereich 12x /56-Netzwerke → 3072 Netzwerke
  - als Reserve

für Gebäude 2:

- für die Infrastruktur 4x /56-Netzwerke → 1024 Netzwerke
  - z.B. für Clients, Drucker, ...
- freigehaltener Bereich 12x /56-Netzwerke → 3072 Netzwerke
  - als Reserve

so weitere Gebäude anschließen

ev. weitere /48-Netzwerke für eine funktionelle Gliederung nutzen  
z.B. Office, Produktion, Infradstruktur, Datacenter, Security, ...

für sehr kleine Struktur-Einheitenn mit sehr wenigen Clients werden besser kleinere Netzwerke zugordnet

- z.B.: /60 → 8 Netzwerke
- /56 → 256 Netzwerke
- /62 → 4 Netzwerke

geeignet für: Arbeitsplätze, VoIP, Sicherheits-Kamera's, Monitore, Zahlstationen / Kassen/ Geldautomaten, ...

nach Funktion getrennte Netzwerke lassen sich einfacher administrieren

immer großzügig vergeben / Reserven lassen

besser agregierbar

besser für Regeln in Firewall

QoS einstellbar

große Netzwerk-Sicherheit

### IPv6-Header

	4	8	12	16	20	24	28	32
	Version	Traffic Class		Flow Label				
	Payload Length			Next Header		Hop Limit		
	Source Address							
	Destination Address							

### IPv6 Header Chain

normales Paket:

NH 6 Fragment Header	TCP Header	TCP Segment
	IP Data	

NH ... Next Header; z.B. NH 6 für TCP

NH 0	NH 44	NH 6	TCP Header	TCP Segment
IP Header	Hob by Hop Options Header	Fragment Header	IP Data	

z.B. IPsec AH Header

	4	8	12	16	20	24	28	32
	Next Header		Payload Length		Reserved			
	Security Parameters Index (SPI)							
	Sequence Number							
	Integrity Check Value (ICV)							
	...							

IPsec ESP Header

	4	8	12	16	20	24	28	32
	Security Parameters Index (SPI)							
	Sequence Number							
	----- Payload Data -----							
	----- Padding (0 – 255 Byte) -----							
					Padding Length		Next Header	
	Integrity Check Value (ICV)							
	...							

Konzept ermöglicht flexibles Protokoll-Design

neue Protokolle / Feature's immer möglich

bedeutet aber auch Probleme beim Filtern ev. problematischer Pakete in der Firewall

IPv6 Fragmentation Header

wenn Pakete zu groß werden, müssen sie aufgeteilt werden

Fragment Header dient zum Teilen von Paketen

	4	8	12	16	20	24	28	32	
	Next Header		Reserved		Fragment Offset			Res.	M
	Identification								

	IPv4	IPv6
<b>Gemeinsamkeiten</b>	Behandlung von zu großen Paketen	
<b>Unterschiede</b>	jedes Gerät auf der Route darf fragmentieren Steuerung über "Do not Fragment"-Bit "DnF"-Bit kann überschrieben	<b>nur Endhost (Absender) darf fragmentieren</b> kein "Do not Fragment"-Bit im Header

	werden Router darf verwerfen  ICMP Packet too Big wird ver- sandt	Router muss zu große Pakete verwerfen ICMPv6 Packet too Big wird ver- sandt
--	-------------------------------------------------------------------------------	--------------------------------------------------------------------------------------

### IPv6-Testlabor

für erste Versuche ältere Hardware nutzen  
 alle typischen Endgeräte-Typen integrieren  
 dann auf aktuelle Hardware wechseln, um alle möglichen Feature's zu testen / nutzen  
 viele verschiedene Client-System nutzen (Win, Linux, Mac, iOS, android, ...)  
 alternativ virtualisiertes System

Umsetzung der Umstellung / Neugestaltung eines IPv6-Netzes

- DMZ
  - Probleme bei: Website's, eMail's, VPN-Verbindungen
  - Dualstack (IPv4 + IPv6) verwenden, damit kein Nutzer abgewiesen wird
  - für klassische Server-to-Server-Protokolle / - Kommunikation IPv6 vorziehen (ev. IPv4 gleich weglassen)
  - Client-Server-Kommunikation immer beide IP-Versionen
  - Virens Scanner muss IPv6-fähig sein
- restliches Netz / Produktion / ...
  - Core Router
  - Datacenter
  - LAN
  - ...

diese Reihenfolge wird Outside-IN genannt (ist immer großes Projekt!)  
 alternativ Inside-Out über die Umsetzung in neuen Abteilungen / Segmenten / ...  
 ev. auch da, wo Umsetzungsdruck am größten ist  
 Inside-Out meist unproblematischer / kleinschrittiger / einfacher / Kosten-günstiger  
 aber ständige Kontrolle notwendig, ob alle Bereiche erreicht werden

### Groß planen – klein starten!

möglichst gleich auf IPv6-only umsteigen, um doppelte Arbeiten an Servern, Firewall's, Rou-  
 tern usw. zu vermeiden  
 ansonsten vielleicht doppelt Migrations-Kosten (IPv4 → Dualstack → IPv6)

### IPv6 und das IoT

#### Wo ist heute schon IoT?

- Heimgeräte (Kühlschränke, Herde, Waschmaschinen, Lampen, Türen, Fenster, ...)
- Kamera's
- Verkehr (Auto's, Ampeln, Laternen, Verkehrsschilder, Verkehrssteuerung, ...)
- medizinische Geräte (→ Dokumentation, Alarmierung, ...)
- Drohnen
- Umwelt- und Produktions-Sensoren
- Roboter, Hochregallager, Gabelstapler, ...

- autonome Systeme

**Probleme:**

- IoT-Komponenten sind meist kleine, Leistungs-schwache Geräte
- Geräte sollen preiswert sein (Sparen z.B. bei der Sicherheit)
- meist keine Update's verfügbar / einspielbar
- allgemein liegen wenig Erfahrungen vor

UDP

HTTP	IMAP	SMTP	...	DNS	...
TCP				UDP	
IPv4					...
ARP (Address Resolution Protocol)					
Ethernet		Token-Ring		...	

→ KALDEALI → S. 350 ff.

TCP

HTTP	IMAP	SMTP	...	DNS	...
TCP				UDP	
IPv4					...
ARP (Address Resolution Protocol)					
Ethernet		Token-Ring		...	

→ KALDEALI → S. 352 ff.



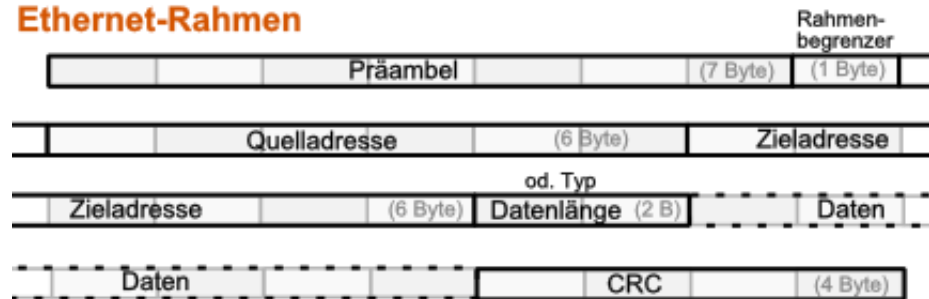
## Übertragungsrahmen Ethernet

→ KALDEALI → S. 313 ff.

alle Daten werden im Ethernet in Daten-Pakete bestimmter Länge zerlegt und in dieser Form übertragen – universelles Übertragungssystem

Verfahren kann bzw. muss jeder Server können, Minimal-Vermögen eines Servers / Relais-Rechners

### Ethernet-Rahmen



dadurch muss nicht jeder Server jedes Protokoll verstehen, er leitet die Pakete einfach weiter, es sei denn sie sind für ihn selbst bestimmt  
nur wenige Protokolle sind quasi verbindlich für alle Server  
z.B. ping und trace

Präambel sind 01010101-Sequenzen, dienen der Synchronisierung  
Adressen sind MAC-Adressen

Länge des Datenfeldes meist vom Protokoll-Typ abhängig, üblich sind 0 bis 1'518 Byte

---

## wichtige Internet-Protokolle

### *electronic Mail (eMail)*

eine der ersten praktischen Nutzungen des Internets  
eines der ältesten Protokolle  
veraltet  
so stark verbreitet, dass kaum noch Änderungen durchsetzbar sind

### POP / IMAP

POP: holt eMails vom Server ab (dort werden sie gelöscht)  
IMAP: holt zuerst einmal nur Kopfzeilen ab, bei Bedarf werden restliche Nachrichten nachgeholt, Nachrichten werden (auf Wunsch) dauerhaft auf Server gespeichert, dort verwaltet

sehr unterschiedliche Client-Programme können alle die Mails abholen, einfache Umsetzung möglich

### SMTP

kopiert eigene (ausgehende) Mails auf den Server; der organisiert dann Zustellung  
sendet eMails an (Mail-)Server

aktuelle Probleme  
historisch / auf Protokoll-Ebene komplett unverschlüsselt und ohne Daten-Schutz  
Spam als Konsequenz

## *Hypertext-System*

### Hypertext Transport Protocol (http, HTTP)

Übertragung von Texten in ansprechender / gestalteter Form sowie Übertragung von Dokumenten und Multimedia-Dateien

Seitenbeschreibungs-Sprache HTML (Hypertext Markup Language)

Text können gestaltet werden  
Positionierung von Verknüpfungen (Links) zu anderen Dokumenten und / oder Dateien  
jede Date erhält / besitzt Adresse → URL (Uniform Resource Locator)

<Protokoll>://<Host>.{<Subdomain>}<Domain>[:<Port>]/<Pfad>/<Dateiname>

neben statischen Seiten und Dokumenten lassen sich auch dynamische Strukturen aufbauen (Datenbank-gestützte Web-Applikationen)

---

**Datei-Übertragungs-Protokolle**

**File Transfer Protocol (FTP)**



---

## 4.x. Clouds – Arbeiten in der Wolke

Bereitstellung von Programmen (Apps) oder Systemen über das Internet oder das eigene Netz

Host ist ein Internet-Server

<https://owncloud.org>

bietet diverse Nutz-Leistungen

- für eigene Rechner nutzbar (z.B. auch auf einem Raspberry Pi)  
viele Cloud-Anbieter nutzen owncloud für ihr Angebot

auch eine virtuelle Maschine für virtualBox vorhanden  
(eigener Server im (virtuellem) Netz)

### **Typen des Cloud-Computing**

<ul style="list-style-type: none"><li>• <b>Infrastructure as a Service</b></li></ul>	<b>IaaS</b> Online-Bereitstellung von Computer-Hardware, z.B. virtuelle Maschinen, Speicher, Netzwerk-Komponenten
<ul style="list-style-type: none"><li>• <b>Platform as aService</b></li></ul>	<b>PaaS</b> Bereitstellung einer Cloud-basierten Umgebung für das Programmieren und Bereitstellen von App's z.B. Systeme der Künstlichen Intelligenz
<ul style="list-style-type: none"><li>• <b>Software as a Service</b></li></ul>	<b>SaaS</b> Online-Bereitstellung von App's und Datenbanken z.B. Nutzung von Office-Programmen oder Warenwirtschaftssystemen (/ von Rechen-Leistung) auf dem Systemen des Anbieter's (im Abonnement)

---

## **4.x. IoT – Internet of Things**

kleine Device's mit geringer Daten-Verarbeitungs-Kapazität  
hauptsächlich intelligente Sensoren und Aktoren mit hohen Kommunikations-Fähigkeiten  
(WLAN, LAN, Internet)

geschätztes Potential: 26 – 50 Mrd. Geräte bis 2020  
durch IPv6 sind wirklich alle Geräte ins Internet direkt anschließbar und ansprechbar (vorher  
(bei IPv4) nur indirekte Anbindung über Cloud's)  
im privaten und technischen Bereich wird starke Vernetzung gewünscht und gefordert; Be-  
reitschaft zur ständigen Bereitschaft  
dazu kommen Smartwatches und Datenbrillen  
Unabhängigkeit vom Smartphone wohl nur eine Frage der Zeit

### ***Herausforderungen an IoT***

- **hohe Interoperabilität** Kommunikation über Protokoll-Grenzen hinweg  
notwendig  
vollständig Hinwendung zu offenen Protokollen
- **hohe Skalierbarkeit** Verringerung der Komplexität
- **breite Kommunikation** Kommunikation zu jeder Zeit und an jedem Ort  
schnell und ständig aktuell  
effektive (Bandbreiten-schonende) Kommunikati-  
on
- **großes Daten-Aufkommen** Big Data
-

---

## 4.x.y. MQTT – das Protokoll für IoT

verwendet wird vorrangig das IoT-Protokoll MQTT (Message Queue Telemetry Transport) seit 2013 standardisiert

alte Versionen (WebSphere MQTT → WMQTT; SCADA-Protokoll; MQ Integrator SCADA Device protocol → MQIsdp)

gibt es für TCP/IP-Netze sowie andere Netze (ZigBee (fünfschichtige ISO-OSI-Umsetzung))

ursprünglich für die Überwachung von Öl-Pipelines von IBM und Arcom Control Systems entwickelt; Nutzung von (teurer) Satelliten-Kommunikation  
aktuell ein offenes Protokoll (seit 2010) unter freier Lizenz  
es gibt auch OpenSource-Implementierungen

Beobachter-Netzwerk mit geringem Verwaltungsdatenteil  
hauptsächlich für Sensoren, aber auch Aktoren, Mobilfunk-Geräte und Embedded Systems (eingebettete Systeme)

möglich Verschlüsselung mit TLS-Protokoll (SSL-Protokoll)

Art der Nachrichten nicht eingeschränkt (z.B. Text, Bilder, Binär-Daten möglich)

### **Vorteile:**

minimaler Protokoll-Overhead

hoch-skalierbar

viele Möglichkeiten

auch für Geräte mit knappen Ressourcen geeignet

Clients für viele Programmiersprachen verfügbar

einfach zu implementieren

schlankes Protokoll (geringer Protokoll-Overhead)

mehrere QoS-Level

es ist Daten-agnostisch (Art der zu übertragenden Daten ist nicht beschränkt)

Kosten-effiziente Daten-Übertragung

### **Nachteile:**

reine Request/Response-Architekturen sind nur mit zusätzlichem Aufwand möglich

typische Nutzungs-Felder:

Instant Messaging / Chat's

Connected Car

M2M-Kommunikation

Ereignis-gesteuerte Publish/subscribe-Architektur (statt Request/Response-System)

Punkt-zu-Punkt-Verbindungen werden durch zentralen Server – dem Broker – ersetzt

Daten-Produzenten (z.B. Sensoren) können genauso mit dem Broker kommunizieren, wie die Daten-Nutzer (z.B. Mobile Endgeräte, Aktoren, ...)

Ein Senden (publish) und Empfangen (subscribe) erfolgt über Strings – Topic's genannt.

der Aufbau eines Topic ist einer URL ähnlich; codiert somit einen Ort für eine Information z.B.

---

schule/haus\_a/raum3012

unter einem Topic (quasi ein Betreff) wird eine konkrete Information (z.B. die Temperatur) abgespeichert

z.B.

schule/haus\_a/raum3012/Temperatur

ein Daten-Nutzer kann genau diesen Topic abonnieren

MQTT-Broker ist für die Verteilung der Daten verantwortlich

Client müssen also nicht ständig den Server über Daten-Änderungen befragen

Daten-Quelle und Daten-Senke wissen nichts voneinander

in Topic's sind Joker (Wildcards) zulässig

+ steht für eine variable Hierarchie-Stufe

# kann mehrere Hierarchie-Stufen umfassen (muss jeweils auch das Ende einnehmen)

richtig: +/+Temperatur (falsch wäre: #/Temperatur)

es gibt drei Service-Qualitäten für die Daten-Übertragung

je höher der Level, umso größer die benötigte Bandbreite im Netz

### **Service-Qualitäten für die Datenübertragung**

- **0** Level 0 – at most once delivery  
("Fire and Forget"-Prinzip)  
keine Zusicherung für Daten-Übertragung  
keine Garantie, das die Nachricht einmal ankommt
- **1** Level 1 – at least once delivery  
( )  
Nachricht kommt mindestens 1x garantiert an
- **2** Level 2 – exactly once delivery  
( )  
Nachricht kommt garantiert nur 1x an

Verfahren zum Erkennen, ob überhaupt eine Quelle vorhanden ist

Verfahren zum Erkennen, ob eine Quelle ausgefallen ist

Protokoll-Konzept "Last Will and Testament" (LWT, Last Will Testament)

Client gibt beim Verbinden mit dem Broker eine Nachricht an, die dann weiterverteilt wird, wenn zum besagtem Client keine Verbindung mehr besteht

Protokoll-Konzept "Retained Message"

eine vom Broker gespeicherte Nachricht, die an jeden Client gesendet wird, der sich neu verbindet

das könnte z.B. der letzte gültige Messwert sein

je Topic ist nur eine Nachricht zulässig

z.B. dann sinnvoll, wenn Daten nur in etwas längeren zeiträumen aktualisiert werden, dann müsste ein neuer Subscriber solange warten, bis der Publisher einen neue Nachricht zur verfügung stellt (die letzte verfügbare Information kann somit sofort angezeigt werden)



---

verschiedene Broker verfügbar

### **Broker für MQTT**

- **Mosquitto**                    Quellen-offen  
                                  kleinere Projekte (z.B. Hausautomation, ...)  
                                  für kleine / kleinste Rechner
- **HiveMQ**                       für anspruchsvolle Telemetrie- und Messaging-Anwendungen  
                                  hoch-skalierbar
- 

Implementierungen / Libery's für diverse Programmier-Sprachen verfügbar

Liste unter: <https://github.com/mqtt/mqtt.github.io/wiki/libraries>

z.B. auch Python, PROLOG, C, C++, Haskell, Jva, JavaScript, Lua, Go, ...

auch für Einplatinen-Geräte wie Arduino / Genuino und Raspberry Pi verfügbar

Client kennt nur die folgenden Methoden

- connect
- disconnect
- subscribe
- unsubscribe
- publish

Arbeitsschritte eines MQTT-Clients

- Erzeugen eines (MQTT-)Client-Objektes
- Setzen der Verbindungs-Optionen ("Last Will")
- Registrieren der Callback's
- ev. Abonieren von Topic's (subscribe)
- Veröffentlichen von Nachrichten (publish)

### Code-Beispiel für Java:

<pre>MqttClient client = new MqttClient( "tcp://broker.mqttdashboard.com, "MyfirstMQTTClient", new MemoryPersistence() );  client.connect();</pre>	
<pre>MqttClient client = new MqttClient( "tcp://broker.mqttdashboard.com, "MyfirstMQTTClient", new MemoryPersistence() );  MqttConnectOptions mqttConnectOptions;  mqttConnectOptions = new MqttConnectOptions(); mqttConnectOptions.setWill( "Zuhause/Wohnzimmer/Temperatur/Status", // Topic "offline".getBytes("UTF-8"), // Nachricht 1, // QoS true); // Retained Message  client.connect(mqttConnectOptions);</pre>	etwas aufwändigerer Verbindungsaufbau mit "Last Will and Testament"
<pre>client.publish( "Zuhause/Wohnzimmer/Temperatur", //Topic "23.4".getBytes("UTF-8"), //Nachricht 1, //QoS true); //Retained Message</pre>	
<pre>client.connect();  client.setCallback(new MqttCallback() { @Override public void messageArrived (String arg0, MqttMessage arg1) throws Exception { // Weiterverarbeiten der Nachricht }  @Override public void deliveryComplete(IMqttDeliveryToken arg0) { }  @Override public void connectionLost(Throwable arg0) { } });  client.subscribe("Zuhause/Wohnzimmer/Temperatur", 2);</pre>	Daten-Nutzer

Q: <https://www.heise.de/developer/artikel/MQTT-Protokoll-fuer-das-Internet-der-Dinge-2168152.html?artikelseite=2>

GitHub-Repository (<https://github.com/dc-square/mqtt-with-paho-eclipsecon2013>) mit weiteren Beispielen

---

Potential für Daten-Übertragung  
zwischen vernetzten Fahrzeugen  
innerhalb von sozialen Netzwerken (Facebook Messenger)  
mit Skalierungs-Möglichkeiten auf sehr hohe Anzahlen gleichzeitiger Verbindungen  
Bereitstellung von Live-Daten (z.B. Patienten-Daten)

Gefahr der vollständigen Überwachung  
Negativ-Beispiel: Zahnbürste "Kolibree", die das Putz-Verhalten u.a. an den Zahnarzt meldet  
(<https://www.kolibree.com/en/>)

**Quelle für diesen Abschnitt:**

<https://www.heise.de/developer/artikel/MQTT-Protokoll-fuer-das-Internet-der-Dinge-2168152.html>

**weitere interessante Links:**

mqtt.org  
[www.eclipse.org/paho/](http://www.eclipse.org/paho/)

---

## 4.x.y.z. alternative Protokolle für IoT-Anwendungen

### **alternative / weitere IoT-Protokolle**

- http
- CoAP
- XMPP
- iBeacon

### **http als IoT-Protokoll**

über http RESTful API's

durch hohe Frame-Zahlen sind schnelle Anzeigen / Reaktionen möglich

Vorteile: stabile und recht leichte Programmierung in fast allen Programmiersprachen machbar, da entsprechende Bibliotheken verfügbar sind

Nutzer-freundlich: Text-basiert, leicht lernbar

nachteilig ist ein recht großer Protokoll-Overhead

nur für 1 : 1-Kommunikationen geeignet

nicht für die 1 : n-Datenübertragung an viele Nutzer

gut geeignet für Downloads großer Daten-Mengen; IoT-Anwendungen mit stabiler Internet-Verbindung und mittelmäßiger Auffrischungs-Raten

### **CoAP**

Constrained Application Protocol

UDP-basiert (auch http für IoT genannt)

sehr effizient

Daten-Übertragung komplett binär

leichter Umstieg von http nach CoAP möglich

Nachteile: komplizierte Network Address Translation (NAT) durch UDP  
wenige Implementierungen (geringere Verbreitung)

gut für Wireless Sensor Networks

---

## **XMPP**

Extensible Messaging and Presence Protocol (früher: Jabber)  
Protokoll wird in Chat's und Instant Messaging-System genutzt  
XML-basiert

Clients für viele Programmiersprachen verfügbar  
viele Möglichkeiten (allerdings hauptsächlich für IM-Anwendungen)

großer Protokoll-Overhead

sehr gut für IM und Chat geeignet

## **iBeacon**

einseitig, da nur Informationen versendet werden (Beacon – Leuchtfener)  
für Informations-Systeme interessant

Nutzung als Informations- und Werbe-Mittel im Handel (Information zu Läden, Gastronomie,  
Sonder-Angebote, ...)

---

## **4.x. IoV – Internet of Value / Blockchain-Technologie**

Quelle: Abschnitt basiert stark auf OpenHPI-Kurs "Blockchain: Hype oder Innovation?"  
von Prof. Dr. Ch. MEINEL und T. GAYVORONSKAYA

### **Problem-Fragen für Selbstorganisiertes Lernen**

Ist ein Handel zwischen Partners sicher möglich, obwohl sich die Partner nicht kennen und potentiell auch (noch) nicht vertrauen?

Wie funktionieren Blockchain-Systeme?

Warum braucht man eine Blockchain-Technologie?

Welchen realen Wert hat ein Bitcoin?

Kann man mit Bitcoin's was kaufen?

Steigt der Wert von Bitcoins ins Unendliche?

Ist die Blockchain-Technologie die moderne Allzweckwaffe gegen Internet- und Realwelt-Betrüger?

Realisiert Blockchain die absolute Wahrheit?

Was schürfen die Miner in Blockchain-Systemen?

Wie kann man das Prinzip der Blockchain-Technologie selber programmieren?

Internet der Werte

es werden z.B. Geld, Objekte, Immobilien, Veranstaltungen, Vermietungen, ... verwaltet

Blockchain → Block-Kette

Blöcke bilden eine Kette, wobei jeder Block einen kryptographischen Verweis auf den Vorgängerblock enthält.

Mit den Transaktionen werden bestimmte Werte von einer Adresse an eine andere übermittelt.

ohne zentrale Administration oder Instanz

jeder soll(te) die Korrektheit der Daten / Identitäten und Veränderungen (Transaktionen)

### **Definition(en): Blockchain**

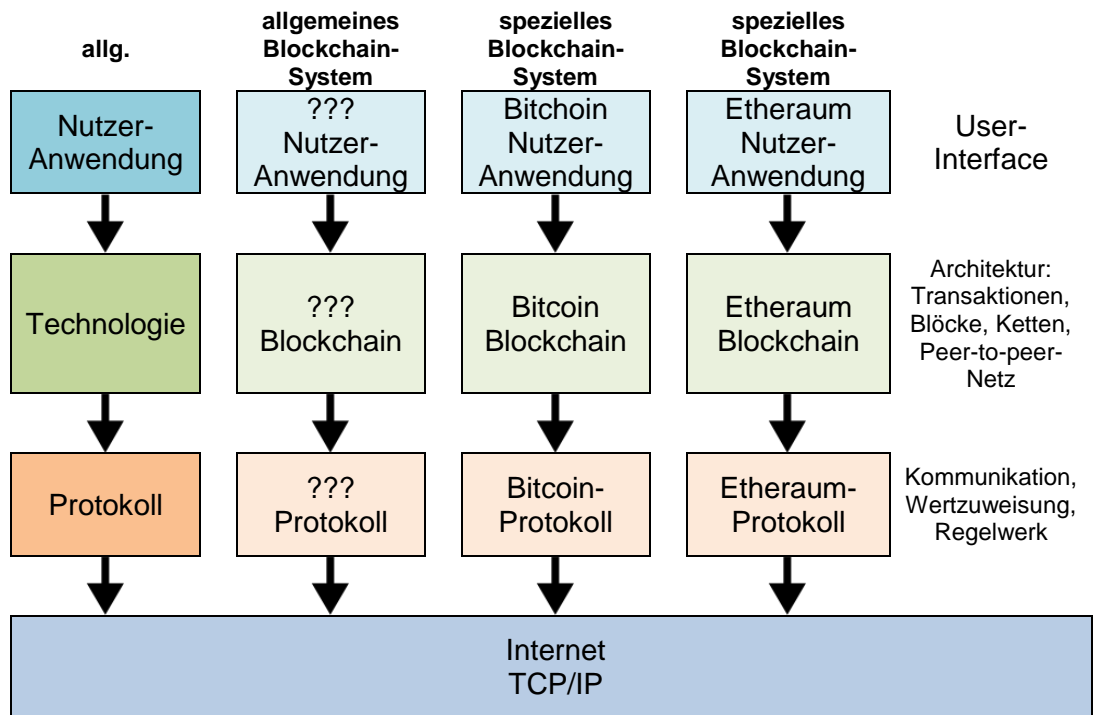
Blockchain ist die Liste aller Transaktionen, die in einem System durchgeführt werden / worden und diese in Blöcke aufgeteilt sind.

Blockchain ist eine Technologie

Bitcoin ist ein System, das auf der Technologie des Blockchain basiert

Möglichkeiten und Tragweite der Technologie noch nicht vollständig abschätzbar

## Schicht-Modell



keine neuen Technologien  
sondern nur neue Kombination der bekannten und etablierten Technologien  
Peer-to-peer-Netzwerke  
Kryptographie

nun neue Konsensfindung im Handeln von Partner, die sich nicht kennen / ? vertrauen

Probleme:

Wer hat welche Zahlung wann (in welcher Reihenfolge) getätigt?  
Wie kann man sich gegenseitig vertrauen?

spezielles Regelwerk für Transaktionen notwendig

Public-Key-Kryptographie

A und B besitzen jeweils für sich ein Schlüsselpaar, bestehend aus einem öffentlichen und einem privaten (geheimen; nur persönlich bekannten) Schlüssel  
A benutzt den öffentlichen Schlüssel von B, um eine verschlüsselte Nachricht an B zu schicken, nur dieser kann die Nachricht mit seinem privaten Schlüssel dechiffrieren

Signaturen zum Prüfen der Herkunft / Quelle / Urheberschaft einer Nachricht / Transaktion basiert auf Public-Key-Kryptographie

Nachricht an sich wird häufig auch damit verschlüsselt, die ist aber für eine Signierung nicht notwendig, es geht nur darum die Originalität / Authentizität der Nachricht zu überprüfen / zu gewähren

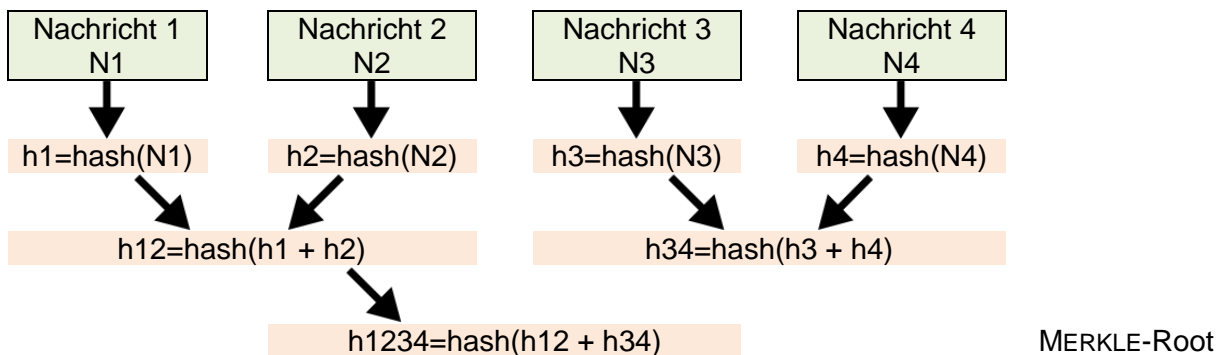
Nachricht oder ein Hash-Wert von einer Nachricht wird mit dem privaten Schlüssel von A verschlüsselt (→ signiert)  
 B kann nun mit dem öffentlichen Schlüssel von A die Nachricht / den Hash-Wert entschlüsseln (→ überprüfen der Signatur)  
 nur wenn sich die Nachricht entschlüsseln lässt, ist sie authentisch / wirklich vom Absender (hier A)

**Aufgaben:**

- 1.
2. ***Vergleichen Sie ausgewogen Verschlüsseln und Signieren einer Nachricht anhand selbstgewählter Kriterien!***
- 3.

um nicht ganze / lange Nachrichten vollständig verschlüsseln zu müssen, verwendet man Hash-Werte über die Nachricht  
 kann man sich wie Finger-Abdrücke einer Nachricht vorstellen; basieren auf Prüfsummen od. ähnlichen eindeutigen Verfahren / Funktionen, die nur in eine Richtung funktionieren  
 Beispiel sin()-Funktion: für jeden beliebigen Winkel lässt sich ein sin-Wert ermitteln, aus einem sin-Wert kann man aber nicht den ursprünglichen Winkel eindeutig zurückableiten, es gibt theoretisch unendlich viele Winkel, die passen könnten  
 (natürlich könnte man bei kleinen Nachrichten-Längen auch auf eine kleinere Winkel-Spanne tippen und diese ev. ausprobieren (→ "Brute force"-Angriff)  
 Hash-Funktionen müssen nun also ebenfalls eindeutige Fingerabdrücke erzeugen und gleichzeitig sicherstellen, dass keine Rückableitung möglich ist  
 Finger-Abdrücke verschiedener Nachrichten sollten auch nur möglichst selten / gar nicht identisch / gleich sein

über einen Hash-Baum lässt sich die Reihenfolge rekonstruieren



**Hash-Baum / MERKLE-Baum**

bei Blockchain wird auf "SHA 256" gesetzt  
 die 256 steht für die Länge des berechneten Hash-Wertes (→ 256 Byte)

über Hash-Werte und die Hash-Bäume lässt sich eine Referenz von Transaktionen / Blöcken erzeugen

Hash-Bäume sind Verallgemeinerungen von Hash-Ketten / Hash-Listen

jeder Nutzer hat gleiche Datenbank



---

## Erklärung der Blockchain-Technologie an ihrer ersten Anwendung "Bitcoin"

die Bitcoin sowie deren Veränderungen werden als Transaktionen ausgeführt und in Blöcken gespeichert

(statt Bitcoin's können aber auch andere Werte / Ereignisse / Besitzverhältnisse verarbeitet werden → im Bitcoin-System eben Geldwerte in BTC)

die Blöcke sind eine Kette aller bisher durchgeführten Transaktionen; jeder Block enthält Referenzen auf andere Blöcke in der Liste

dieser Datenbestand steht allen Nutzern zur Verfügung

durch die Anwendung der Kryptographie ist weder eine Veränderung der Blöcke (Transaktionen) noch der Verkettung der Blöcke untereinander möglich

### Transaktion

kleine technische Einheit der Blockchain-Technologie besteht aus Input und Output

der Input enthält die Referenz auf frühere Transaktionen

daraus ergibt sich der aktuelle Kontostand, also wieviele Werte

im Output stehen die initiierten Zahlungen (ausstehende Transaktionen)

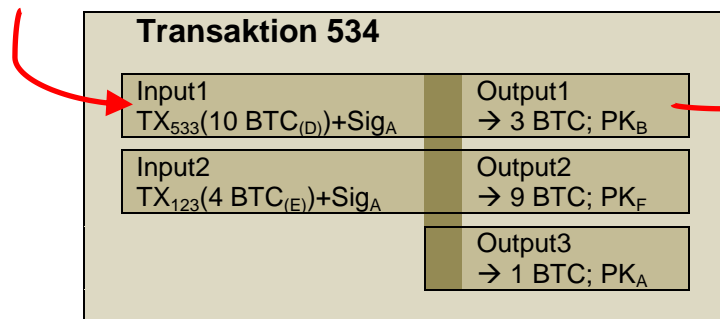
Transaktion 534 ( $TX_{534}$ )

besteht aus mehreren Teilen. Da sind zwei Inputs. Der eine Input folgt aus der Transaktion 533 und ist mit einem "Zahlungseingang" von 10 BTC von Nutzer D verbunden. Um sicher zu stellen, dass die "Zahlung" beim richtigen Nutzer A ankommt, ist diese mit der Signatur (Public Key) von A verschlüsselt.

Ein zweiter Input erfolgt aus der Transaktion 123.

Die Outputs sind in diesem Fall drei-geteilt. Eine "Überweisung" erfolgt an Nutzer B. Unter Verwendung dessen öffentlichen Schlüssels (PK) werden 3 BTC transferiert.

vorlaufende Transaktion  
(dieser Kette)



folgende Transaktion  
(dieser Kette)

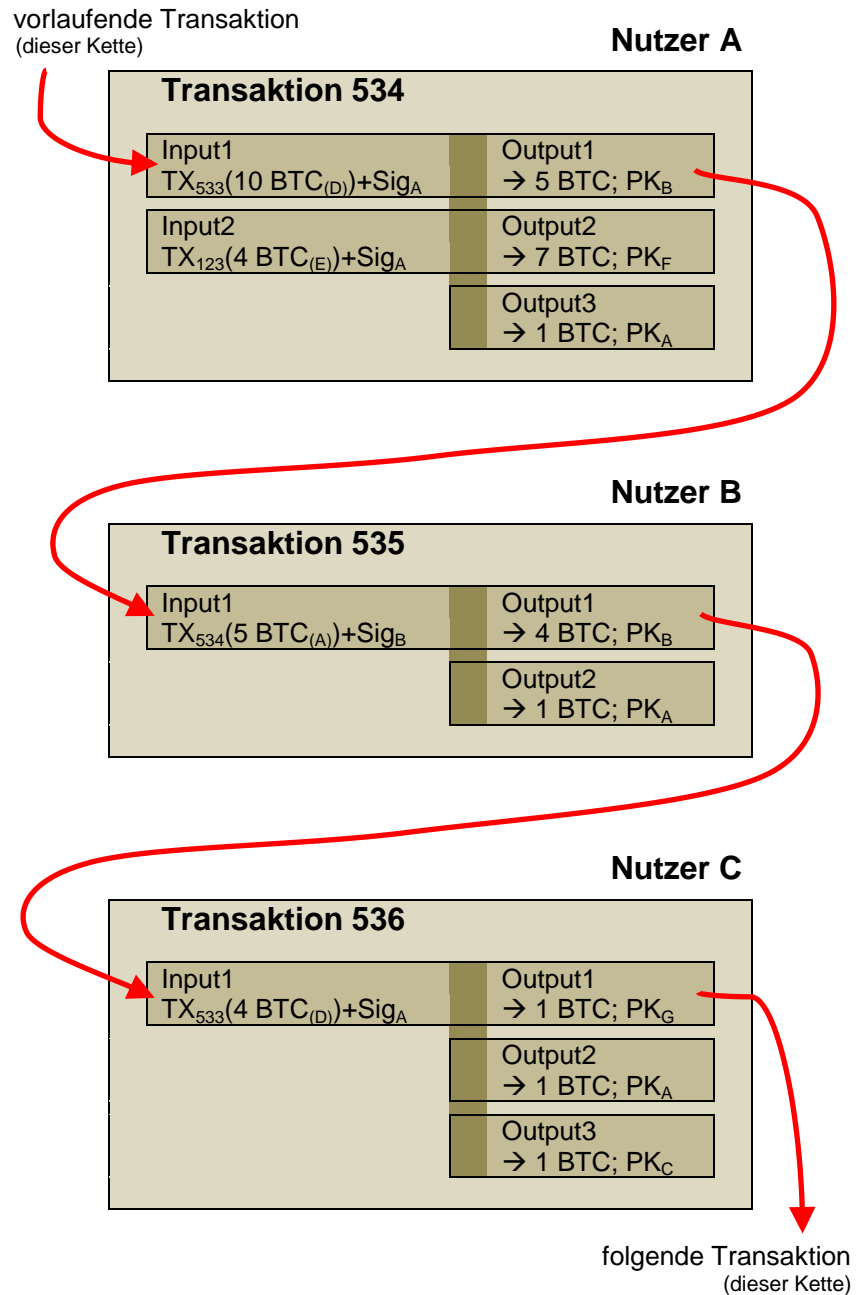
### Aufgaben:

1. Welchen Inhalt hat der Output2? Erläutern Sie!
2. Beschreiben Sie die Überweisung aus Transaktion 123 in ihren Details!
- 3.

Der dritte Output ist eine System-interne Überweisung an sich selbst. Dabei wird der Restbetrag (nach den anderen Überweisungen) abzüglich einer Transaktions-Gebühr (hier 1 BTC) übertragen.

Betrachtet man nun eine Liste von Transaktionen, so enthalten diese alle notwendigen Angaben, um die Transaktion in der richtigen Folge zu rekonstruieren und deren Echtheit zu überprüfen.

bei den Transaktionen sind Transaktionsgebühren fällig  
 deshalb ist es notwendig den Restbetrag zwischen Input und Abgängen (Überweisungen + Gebühr) an sich selbst zu überweisen, sonst gehen die Differenz in der Gebühr unter



Alle Transaktionen werden an alle Nutzer des Systems verschickt und stehen lokal zur Verfügung. Dadurch ist eine nachträgliche Manipulation nicht mehr möglich.

Jeder Nutzer prüft die Daten in den Blöcken und speichert sie in der lokalen Datenbank ein.

---

Prüfung umfasst die Signatur der Transaktion, die Einmaligkeit der Transaktion und den Empfänger (der Transaktion). Ist man selbst der Empfänger, dann wird die Transaktion in die eigene Wallet eingearbeitet.

Die Miner sind die "zentralen" Verarbeiter der Transaktionen zu Blöcken. Da diese Arbeit sehr rechenaufwändig ist, wird sie belohnt z.B. Bitcoin's. So entstehen also auch neue Bitcoin's.

Blöcke enthalten:

Header: Zeitstempel, Software-Version, Hash des vorherigen Blocks (Block-Header + Nonce()), Mining-Angaben (Nonce, Zielvorgabe); Transaktions-Anzahl, MERKLE-Root (256 bit)

Body: Liste der Transaktionen

Im Bitcoin-System wird mit zwei unabhängigen Hash-Werten gearbeitet.

Beim Erstellen der Blöcke arbeiten u.U. mehrere Miner an der Block-Bildung über die gleichen oder ähnliche viele Transaktionen. Alle Miner verteilen ihre berechneten Blöcke im System.

Treten jetzt scheinbar parallele Blöcke auf, dann werden zuerst einmal Seiten-Ketten (Fork (Verzweigung / Gabel)) genannt in der Kette angelegt. Sind Blöcke gleich, dann wird mit einem Zweig weitergemacht. Es entsteht eine verlängerte Haupt-Kette. Der andere Block (aus der Seitenkette (Side??)) kann beruhigt "verworfen" werden. Er verbleibt zur Sicherheit im System. Ein solcher Orphan-Block wird aber nicht mehr weiter beachtet.

Kommen Blöcke mit mehr (beinhalteten) Transaktionen an, dann hat diese Vorrang und der Block mit der kürzeren Transaktions-Kette wird verworfen.

Über Konsens-Regeln werden also die vollständigeren Informationen den geringwertigen vorgezogen.

Nur der erste Miner mit der längsten Block-Kette erhält Belohnung (aus den Transaktions-Gebühren). Die unterlegenen / langsameren Miner verlieren ihre Rechen-Leistung / aufgewendeten Ressourcen.

Regelwerk und Konsensalgorithmen

### **Regelwerks-Bereiche**

- **Architektur**
  - Blockgröße
  - Blockerstellungzeit
  - kryptographische Verfahren
  - Hash-Verfahren
  - kryptographische Verweise / Verknüpfung der Hash's
  
- **Datenverarbeitung**
  - Anzahl der eingehenden Verbindungen
  - Anzahl der ausgehenden Verbindungen
  - DoS-Prävention
  
- **Datenverifizierung**
  - (syntaktische) Korrektheit der Nachrichten
  - Lösch-Regel
  
- **Mining / Minting**  
(Fortschreibung der Blockchain)
  - Proof-of-Work (betrifft Mining; (wahrscheinlich) aufgewendete Ressourcen)
  - Proof-of-Stake (betrifft Minting: Bewertung der Besitztümer; Regeln zum Neubau von Blöcken)
  - Delegated Proof-of-Stake ( )
  - Proof-of-Burn (betrifft Mining: )
  - Stellar Consensus Algorithmen
  - Federated Byzantine Agreement

Minting betrachtet die vorhandenen Ressourcen / Werte / ... / Stimmen anderer Nutzer

---

z.B.: **Proof-of-Work**

kryptographische Aufgabe

es werden Schwierigkeits-Grade (difficulty) festgelegt

Lösen einer (schwierigen) Aufgabe in z.B. 10 min (Berechnung gerade so möglich)" → Ansporn" im System; Aufgabe ist die Berechnung des Hash-Wertes

Die Schwierigkeit liegt darin einen kleinen / passenden Hash-Wert zu erzeugen.

Miner schließen sich zu Pool's zusammen, um das Problem am schnellsten zu lösen.

Gefahr liegt in der Übermacht einzelner Pool's. Da nur die Gleichberechtigung das notwendige Vertrauen garantieren kann.

weitere Aufgaben können unterhalb der Zielvorgabe (difficulty target) erledigt werden

### **Stärken / Vorteile der Blockchain-Technologie**

- **sehr hohe Nachverfolgbarkeit** Problem bei Dominanz einzelner Nutzer / Pool's
- **sehr hohe Fälschungs-Sicherheit** Problem bei Dominanz einzelner Nutzer / Pool's
- **Ausfall-Sicherheit** durch Peer-to-peer-Netzwerk gegeben
- **Autonomie** keine zentrale Instanz notwendig (z.B. Bank)
- **geringe Angreifbarkeit** zu viele Datenbestände an zu vielen Orten
- **starke Kryptographie**
- **OpenSource** breite Entwicklergemeinde  
aber auch Problem bei der Fehlernutzung
-

---

### **Herausforderungen**

- **Datenbestand** z.T. sehr hoch
- **ev. notwendige (partielle) Zentralisierung** zur Reduzierung des Daten-Bestandes beim Client
- **Skalierbarkeit** durch große Daten-Bestände eingeschränkt
- **Haftung / Datenschutz**
- **Interoperabilität mit anderen Systemen** derzeit fehlen Standards
- **weitere Anwendungsszenarien**
- **Angreifbarkeit des Systems** verschiedene Angriffs-Szenarien möglich  
z.B. 51%-Angriff
- **Ressourcen-Verbrauch** z.T. beachtlich, um Sicherheit zu gewähren
- **optimale Position im Trilemma (Dezentralisation, Skalierbarkeit, Sicherheit)** alle drei Parameter des Trilemma lassen sich nicht zusammen maximieren, das geht nur für maximal zwei
- **Kosten-Nutzen-Verhältnis** ev. sind andere Anwendungen / Technologien günstiger / sinnvoller und universeller

---

## Sicherheit in Blockchain-Systemen

### **Angriffs-Szenarien auf Blockchain-Technologien**

- **Denial-of-Service-Angriff** es werden so viele Transaktionen erzeugt, dass das System mit der Block-Berechnung nicht mehr hinterher kommt und dann versagt bzw. manipulierbar wird (über Gebühren-Regeln eingeschränkte Gefahr)
- **Flood-Angriff / Spam-Transaktionen** Angreifer erzeugen viele Transaktionen auf sich selbst (fluten das System)
- **Sybil-Angriff** nach einer psychisch gestörten Roman-Figur benannt es wird mit mehreren Identitäten gearbeitet, die falsche / manipulierte Informationen / Transaktionen versenden Nachbar-Nutzer werden mit Falsch-Informationen versorgt und erzeugen so Disharmonien im Netzwerk / System
- **Verfolgung der Transaktionen** Erforschung der echten Nutzer-Adresse / -Daten teilweise Abhilfe z.B. durch TOR-Netzwerk-Technologie alternativ auch Mixing Service (Verknüpfung der Blockchain-Technologie mit anderen Internet-Services, um die Blockchain-Aktivitäten zu verschleiern (Gefahr z.B. für Geldwäsche, ...))
- **Ausspähen von geheimen Schlüsseln** unabhängig von eigentlicher Blockchain-Technologie social Hacking Gefahr durch Schlüssel-Tresore bzw. Hardware-Wallet's reduziert
- **51%-Angriff** ein Miner / Miner-Pool hat über 50% der Block-Berechnungs-Kapazität (Monopol-Bildung), dann ist Manipulation möglich sowohl Proof-of-Work und Proof-of-Stake sind über 51%-Angriff manipulierbar
- **Probleme / Fehler / Lücken in den verwendeten Technologien** Fehler in Netzwerk-Protokollen / Programmen / ... (vor allem proprietärer Software / Umsetzung / ...)

---

### **Beispiele für geplante / realisierte Blockchain-Systeme**

- **Bitcoin**
- **Ethereum**
- **IOTA** für den IoT-Bereich
- **Ledger-Systeme** elektronische Kontobuch-Führung; DLT
- **eGovernment**
- **Wahl-Systeme** elektronische Stimmabgabe
- **Produkt-Traking**
- **Audit-System** Vorteil: nicht die erfassten Daten müssen gelöscht werden, es reicht das Löschen des privaten Schlüssels
- **elektronische Gesundheitsakte**
- **Militär(-Geheimnisse)** Waffen-Entwicklung  
Aktivierung / Auslösung von Waffensystemen  
Befehls-Ketten
- 
-

---

## 4.x.y. Krypto-Zahlungssysteme

erste Anwendung für Blockchain-Technologie

voll-elektronisch

ohne zentrale Instanz / Administration

kryptographischer Beweis statt Vertrauen ("Geschäftsbasis")

### **4.x.y.0. Grundlagen**

klassische Blockchain-Systeme

kaum Anpassungen notwendig

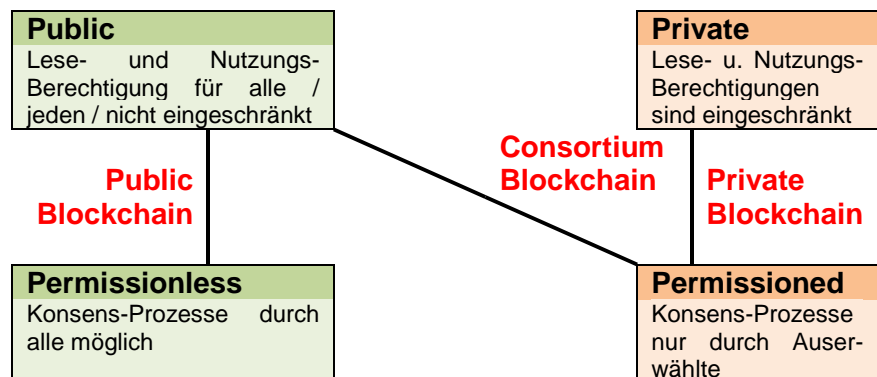
zusätzliche App's / Web-Plattformen, um Nutzer ansprechen zu können



---

## Blockchain-Typen

- **Public Blockchain** alles öffentlich  
jeder darf das System nutzen, Miner sein (realisiert Konsens-Prozess),  
für Kommunikation von Firmen u. / od. Kunden, die sich selbst nicht kennen (wollen) und somit auch noch kein Vertrauens-Verhältnis existiert
- **Private Blockchain** eingeschränkte (private) Nutzung / Anwendung  
nur bestimmte Nutzer dürfen das System nutzen  
nur bestimmte Nutzer / ??? machen Konsens-Prozesse  
z.B. für Firmen-interne Kommunikation
- **Consortium Blockchain** teilweise öffentlich  
jeder darf das System nutzen  
nur bestimmte Nutzer / ??? machen Konsens-Prozesse  
z.B. für Firmen-Kunden-Beziehungen / -Kommunikation



## Kriterien für den Einsatz der Blockchain-Technologie

- **(klares) Ziel** ? Werte und Anwendungsbereich
- **Möglichkeiten und Herausforderungen** ? Transparenz der Blockchain  
? Berechtigungen zum Fortschreiben der Blockchain
- **Kosten-Nutzen-Verhältnis** ? Nutzung bestehender Blockchain's  
? neue Blockchain's
- **Umsetzungs-Möglichkeiten** ? Ressourcen

## Altchain (Alternative (Block-)Chain)

erzeugen einer neuen Blockchain für die eigene Anwendung  
bedarf der Realisierung der gesamten notwendigen Bestandteile (Implementierung der Technologien)

---

## **Forking**

beim Nutzen bestehender Blockchain's wird man mit dem Aufspalten (Forking) der existierenden Blockchain konfrontiert, für die eigene Nutzung wird ein Fork (eine Verzweigung; Neben-kette) aufgemacht  
ursprüngliche Anwendung wird mit ihrer eigenen Blockchain (Hauptkette) weitergeführt  
beim Verifizieren der Fork's wird an der Gabelstelle ein "Merged Mining" benötigt

## **Hard Fork**

hier werden grundsätzliche Änderungen an der Architektur des Blockchain vorgenommen (z.B. Vergrößerung der Block-Größe, ...)  
alle Nutzer sind betroffen und müssen mitmachen / zustimmen

## **Soft Fork**

neue Funktionen; hinzugefügte Anwendung  
hierbei sind nur die Nutzer des Fork's (Blockchain-Zweiges) betroffen  
Miner und Fork-Nutzer müssen sich auf den neuen Fork einstellen und ev. neu agieren

## **Colored Coins**

verwaltete Werte in der Blockchain werden durch zusätzliche Meta-Daten erweitert  
den Daten (Werte) wird ein Typ zugeordnet → als Farbe betrachtet  
die unterschiedlichen Datentypen werden dann aber gemeinsam in der Blockchain verwaltet  
für die Miner sind die Blöcke neutral, die Miner interessiert nicht der Inhalt der Daten, sondern sie realisieren nur die Sicherheit und Konformität / Validität der Kette  
die Anwender von speziellen Colored Coins (Daten-Typen) müssen mit ihren Anwendungen (App's) die für sie betreffenden Transaktionen aus der Kette filtern können  
die anderen Transaktionen ( anderer Colored Coins) sind für sie nicht zugänglich

Colored Coin = Bestehender\_Wert + Metadaten

Nutzer wissen, um welche Werte / ... es sich handelt  
Miner und Minter können mit den Farben / internen Daten nichts anfangen; sie sind praktisch für sie (semantisch) unlesbar

## **Sidechain**

Übertragung von Werten / Transaktionen in andere Ketten  
in der Blockchain werden die Daten anderer Ketten erkannt und geprüft  
Erhöhung der Interoperabilität (zwischen Blockchains)  
Vernetzung der (speziellen) Blockchain mit anderen Blockchains  
z.B. Kauf von Grundstücken (Liegenschaften-Blockchain) mit Bitcoin (Bitcoin-Blockchain)

## **Pegged Sidechain**

Austausch der Werte / Transaktionen ist zwischen beiden Ketten hin und her möglich

---

## 4.x.y.1. Anwendungs-Beispiele zur Blockchain-Technologie

### **Smart Contracts ()**

oft schon als Blockchain 2.0 bezeichnet

es wird mit Cored Coins gearbeitet (z.B. Handel mit Grundstücken / Immobilien)

kein Fork

zusätzliche Ebene / Erweiterung der bestehenden Blockchain-Technologien

??? zwei Zustände: ausgegeben / nicht ausgegeben (z.B. bei Krypto-Währungen) ???

kleine Mengen von Metadaten

neue Transaktionen erfordern es immer, die gesamte Blockchain (der betreffenden Colored Coins) zu prüfen

Fork oder ganze Kette muss geprüft werden

autonomer Agent

unabhängiges Programm (dezentrale App → DApp)

- Programmierung erfolgt in beliebiger Programmiersprache)
- wird in interner Byte-Code-Sprache an alle Nutzer verteilt
- dieser Byte-Code kann von Nutzern dann ausgeführt / genutzt werden)
- Verwendung einer Runtiem-Umgebung bzw. einer virtuellen Maschine zum Ablaufen des Byte-Codes's

besitzt eigene Blockchain-Adresse / Nutzer-Adresse

wird durch Transaktion "aktiviert"; vom Nutzer angesprochen / ausgelöst

Realisierung über Ethereum-System

ab 2014

praktisch Werkzeug-System zum Benutzen der Blockchain-Technologie

Werte-Basis ist die Krypto-Währung Ether

mit Token (Koffer / Kiste)

Token enthält (Werte, Informationen (Daten), Bedingungen / Regeln)

Miner behandeln die Smart Contracts als ihre Daten-Basis; verifizieren diese und bestätigen Echtheit durch Mining (kryptographische Methoden)

Transaktionsliste ist von den Zuständen abgetrennt

Nutzer verfügen (nur) über den letzten Zustand (State, System-Status) des Smart Contract

(bei Bitcoin anders, hier muss erst die gesamte Kette verifiziert werden / ein neuer Block gebildet werden, damit allen Nutzern im System der gültige Zustand übermittelt werden kann)

Zustände und Transaktionen haben eigene MERKLE-Bäume (bei Ethereum heisst dies Merkle Patricia Tree)

es gibt Smart Contracts, die niemals ablaufen / ungültig werden (→ DAO's)

DAO = dezentrale autonome Organisation

Oracle's

Blockchain-Konzept, das Ereignisse / Sachverhalte aus der Realwelt verifiziert und als Smart Contract's bereitstellt

eine Art digitaler Agent

quasi Schnittstelle zwischen Außenwelt (Nutzer; auch Sensoren und Aktoren) und des Eigenlebens der Blockchain

Verbindung zwischen IoV und IoT

verschiedene Arten von Oracle's möglich:

---

### Oracle-Arten

- **Hardware Oracles** direkter Zugriff auf Sensoren (RFID, Mess-Fühler, ...)
- **Software Oracles** online verfügbare Daten (Flugdaten, Verspätungen, Wetterdaten)
- **Inbound Oracles** z.B. Kauf- / Verkauf-Order in Abhängigkeit vom einem bestimmten (Börsen- / Handels-)Kurs oder Preis beziehen sich auf eingehende Informationen / Daten als Kriterium
- **Outbound Oracles** Steuerung von Außenwelt-Elementen (Schlösser, Geräte, ...) in Abhängigkeit von bestätigten Transaktionen, ...
- **Consenses Based Oracles** Kombination von mehreren Arten von Oracle's es wird ein Konsens erarbeitet (z.B. es müssen drei von fünf Oracle's zutreffen / bestätigen, um eine Transaktion / ... auszuführen)

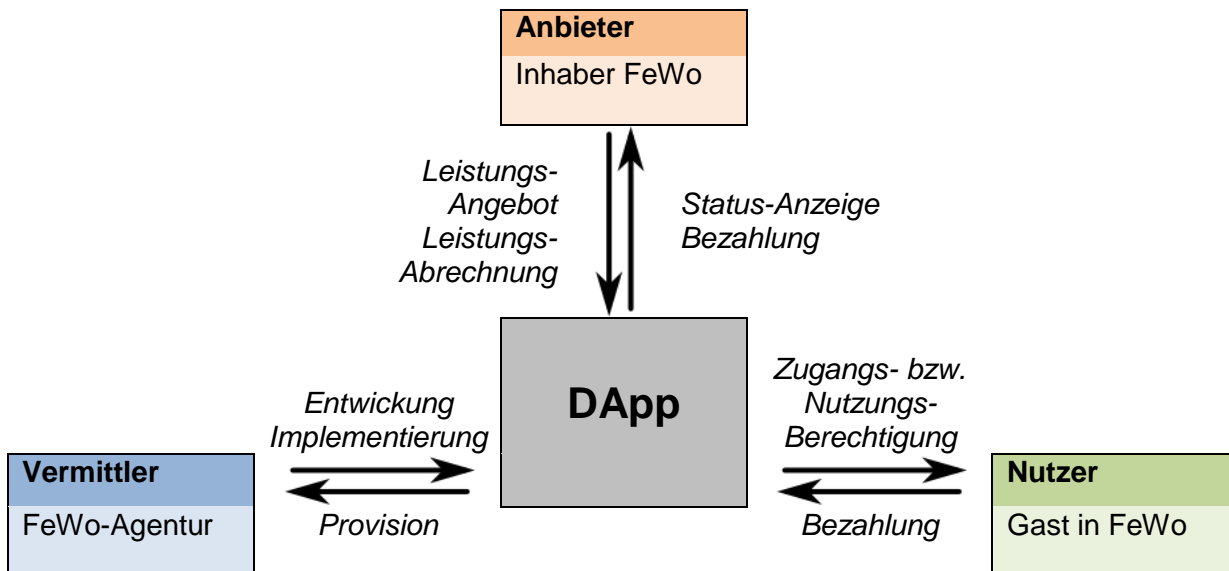
<b>Digital value exchange</b>	<b>Smart right and obligation</b>	<b>Basic smart contract</b>	<b>Multiparty smart contract</b>	<b>Distributed autonomous bussiness unit</b>	<b>Distributed autonomous organisation</b>	<b>Distributed autonomous government</b>	<b>Distributed autonomous society</b>
digitaler Werte-Handel				dezentrale, autonome Firmen	dezentrale, autonome Organisationen	dezentrale, autonome Verwaltung	dezentrale, autonome Gesellschaft / Gemeinschaft
Versand von Bitcoins zwischen Nutzern	Nutzer kauft digitalen Inhalt (als Stream)					Verwaltungs-Angelegenheiten in einer Regierungseinheit	alle Beziehungen z.B. innerhalb einer Stadt oder eines Staates



**Komplexität**

nach Q: blockchainhub.net

## Anwendungs-Beispiel (Idee): Ferienwohnungs-Vermittlung



### (potentielle) Anwendungsbereiche für die Blockchain-Technologie

- **Medizin**
- **Identitäts-Management**
- **Cloud-Computing**
- **Internet of Things (IoT)**
- **Energie-Versorgung**  
**Energie-Management**
- **Buchungen / Vermietungen**
- **Finanzwesen**
- **Logistik**
- **Immobilien-Handel**  
**Immobilien-Management**
- **soziale Netzwerke**  
→ steemit  
→ .publicism  
Leistungs-Abrechnung / -Bewertung von Post's / publizistischen Artikeln / ...
- **Lieferketten / Herkunftsnachweise**  
→ CLEAR Karma  
Nachweise für die Herkunft von Zutaten in Lebensmitteln
-

---

### **Startup-Projekte**

- Storj
- Modum.io
- Slock.it
- Consensys
- Blockstack
- 

### **Firmen, die Blockchain-Projekte nutzen:**

- RWE
- SAP
- Foxconn
- Porsche
- IBM
- Microsoft
- Samsung
- Intel
- Siemens
- 

### **Konsortien, die auf Blockchain-Technologie basieren:**

- Hyperledger
- R3
- Enterprise Ethereum Alliance
- BCCC
- Chain of Things
- 

### **derzeitige strategische Zielrichtungen:**

- Protokollierungen (von Aktivitäten, Transaktionen, Ereignissen, ...)
- verteilte Datenbanken (sichere Daten-Speicherung und –Manipulation)
- State Machine (→ endliche Automaten)

### **Typen von Blockchain-Systemen**

- öffentlich (public) oder privat (private)
- Produkt, Service oder Plattform (Blockchain as a Platform (BaaP); Blockchain as a Service (BaaS))
- Startup's / Kooperationen
- Blockchain als Produkt
- 

### **internationale Blockchain-Projekte:**

- Australien (allgemein)
- Estland (Verwaltung / Protokollierung)
- Schweden (Liegenschafts-Verwaltung)
- Niederland (Modell-Stadt: )
-

---

## **weitere "sinnvolle" Mining-Anwendungen von Blockchain-Technologien**

- **BOINC** lösen von wissenschaftlichen Aufgaben meist internationaler Projekte (verteilttes Rechnen)  
auch als Bildschirmschoner (sehr schönes Mitmachprojekt für Jederman)
  - SETI (Suche nach extraterrestrischen Funksignalen)
  - Einstein (Suche nach Gravitationswellen)
  - Rosetta (Faltung von Proteinen)
  - ...
  
- 
-



---

## 4.x.y.2. Blockchain-Hauptanwendung: Finanzwesen und Krypto-Währungen

2008 von Satoshi NAKAMOTO entwickelt und 2009 als OpenSource-Software veröffentlicht.  
die Person Satoshi NAKAMOTO ist wahrscheinlich ein Pseudonym für eine Entwickler-Gruppe.

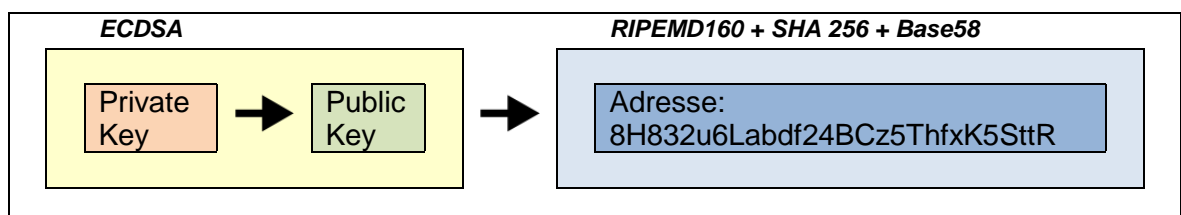
### 4.x.y.2.0. "normale" Bankgeschäfte (ohne Blockchain)

normaler Zahlungs-Verkehr immer über zentralen Partner (die Bank)  
ihr vertrauen Auftraggeber und Empfänger einer Zahlung / Transaktion  
??? Vertrauen (Abhören / Datensammlung / -spuren)  
entspricht eher der Client-Server-Architektur (gesamt Intelligenz / Leistung) liegt auf Server  
(hier: die Bank)

### 4.x.y.2.1. Bankgeschäfte mit Blockchain

Vertrauen wird in der Bitchain-Technologie dadurch erzeugt, dass alle anderen Zeugen von Transaktionen sind / werden

#### Generierung der Nutzer-Adresse



Private Key ist eine sehr große Zufallszahl; aus dieser wird der private Schlüssel abgeleitet / berechnet

jeder Nutzer erhält DNS-Namen (genannt: seeds), mit Hilfe dessen sich die Nutzer ansprechen können (TCP/IP-Adressen der Nutzer)  
seeds.bitcoinstats.com

hardware wallet zur Nutzung des Wallet

---

## Definition(en): Bitcoin

Bitcoin (BTC) ist ein Krypto-Währungssystem, das auf der Basis der Blockchain-Technologie basiert.

eine Beteiligung im Bitcoin-System ist nur möglich, wenn man über eine Bitcoin-Geldbörse (Wallet) verfügt; Nutzer beteiligen sich über ein Pseudonym am System

das sind Programme / Apps

Wallet kann man auch als Anwender-Programm betrachten

beinhaltet "Kontonummer" (Bitcoin-Adresse) und "Kontostand"

kryptographisches Schlüssel-Paar wird für die Anwendung unbedingt gebraucht

mit privatem Schlüssel werden die Transaktionen erstellt und versendet → Verschlüsseln

mit dem öffentlichen Schlüssel wird die Nutzer-Adresse (Bitcoin-Adresse, "Kontonummer") erstellt → Signieren und Bestätigung der Signatur

Erzeugen neuer Bitcoins wird über das sogenannte Mining realisiert. konstanter Zufluss neuer Bitcoins

festgelegte Obergrenze sind 21 Mio BTC, die voraussichtlich 2032 zu 99% erreicht werden  
Obergrenze dient der Verhinderung einer (unendlichen / steigenden) Inflation

### **Herausforderungen**

- **Datenbestand** derzeit (2018) rund 170 Gbyte  
rund 12'000 Nutzer / Knoten
- **ev. notwendige (partielle) Zentralisierung** zur Reduzierung des Datenbestandes beim Client
- **Skalierbarkeit** bedingt durch unterschiedliche Leistungs-Parameter bei den Nutzer-Geräten (SmartPhone's, PC's, Server-Farmen, ...) werden unterschiedliche Nutzerklassen eingeführt:
  - Lightweight nodes und Full nodesunterschiedliche Datenbestände (in einfachen Versionen nur noch Header im Datenbestand → nur noch eigene Transaktionen im Datenbestand  
Vertrauen bezieht sich nur noch auf die selbstbezüglichen Transaktionen  
Trennung von Kleinstzahlung vom "großen" Handel
- **Haftung / Datenschutz**
- **Interoperabilität mit anderen Systemen** derzeit fehlen Standards

- 
- **Angreifbarkeit des Systems** verschiedene Angriffs-Szenarien möglich
  - **Energie-Verbrauch** Bitcoin-System verbraucht pro Jahr 11,2 TWh (1 Atomkraftwerk)  
je Transaktion wird soviel Strom verbraucht, wie 4 Industriestaaten-Haushalte pro Tag  
(Visa-System benötigt 1/10'000 der Energie-Menge)

### ***Krypto-Währungen***

- **Litecoin**
- **Ethereum (Ether)**
- **XRM von Monero**
- **XRM von Ripple**
- **DASH**
- **Gridcoin**
- 

Anwendung in Börsen  
derzeit NASDAQ (USA) und ASX (Australien)  
Clearing von Transaktionen

Banken, die Blockchain-Technologie einsetzen:

- Deutsche Bank
- Santander
- Barclays
- UBS
- 

Einsatz auf verschiedenen Bereichen, außer Transaktionen auch Interaktion mit Kunden und anderen Finanz-Dienstleistern

Nutzung auch beim Crowdfunding (Geld-Sammlung für Projekte)

### ***eingesetzte Technologien (der Blockchain-Technologie) im Bereich Finanzwesen***

- **(klassische) Blockchain** z.B. Krypto-Währungen
- **Colored Coins**
- **Smart Contracts**  
State Machine's (Entscheidungs-Automaten)
- **Oracles**
-

---

***Konsortien, die Blockchain-Technologie benutzen bzw. auf diese basieren***

- **BCCC** Japan  
über 100 Mitglieder
- **r3**  
über 70 Mitglieder
- **Enterprise Ethereum Alliance**  
über 500 Mitglieder
- **Microsoft + ITRI + AMIS**

**Hauptanliegen / Arbeitsbereiche der Konsortien:**

- Schaffung von Standards
- Bereitstellung von Plattformen
- Bereitstellung von Anwendungen (z.B. Finanzwesen)

---

### 4.x.y.3. Anwendung: Cloud's und Cloud-Computing

verteilte Datenbanken

Cloud-Speicher-Projekte

- Store??? (OpenSource; verteilte Cloud-Speicher) arbeitet mit "Farmer"n, die die verschlüsselten Daten speichern und dafür (mit eigener Krypto-Währung) entlohnt werden
- 

Cloud-Computing

- ORACLE Cloud
- amazon web services
- Microsoft Azure
- IBM
-

---

#### **4.x.y.4. Anwendung: Internet of Things (IoT)**

derzeit sehr unterschiedliche Cloud-Infrastrukturen  
noch aktuelles Problem der Nutzung von IPv4 als Netzwerk-Adressen, was (wegen Adress-Mangel) mit einer indirekten Kommunikation verbunden ist  
mit der Umstellung / vollständigen Nutzung von IPv6 ist eine direkte Anbindung von Sensoren und Aktoren an die Steuer-Strukturen möglich

Blockchain-Technologie für sichere, protokollierte und strukturierte Kommunikation eingesetzt

derzeitige Standard eingeschränkt; Interaktion zwischen den Standard eingeschränkt; hier Einsatz von Blockchain-Technologie, um die Daten zwischen den System sicher austauschen zu können

Suche nach übergreifenden / allgemeinem Standard für IoT

Technik soll / muss sicher, robust, protokollierbar und skalierbar sein

##### **Herausforderungen:**

- Rechenleistung (beschränkt / gering)
- Speicherkapazität (beschränkt / gering)
- geringe Akku-Leistung / Energie-Verbrauch
- Skalierbarkeit
- Transparenz
- Sicherheit
- Kombination / Koppelbarkeit von Standard's

##### **Möglichkeiten / Potential:**

- Kosten-Ersparnis
- Ausfall-Sicherheit
- Daten-Integrität

##### **Anwendungs-Beispiel: Slock.it**

arbeitet mit Ethereum-Technologie

deutsches Startup

Einbindung auch in SmartHome-Systeme

Steuerung über Handy-App

##### **Anwendungs-Beispiel: FILAMENT**

Industrie-Bereich

Verfolgung von Bauteilen und Produkten vom Hersteller bis zum Nutzer

setzt auf Bitcoin-System

setzt auf:

- Security (Sicherheit)
- Privacy (Privatheit / )
- Autonomy (Eigenständigkeit)
- Decentralization (Dezentralisation)
- Exchange (Handel)

---

## **chain of things**

deutsches IoT-Konsortium (mit Handlungsbereich Blockchain-Technologie)

Ziel-Bereiche /Focus-Bereiche:

- Chain of Security (für sichere IoT-Anwendungen)
- Chain of Solar (für Solar- und Energie-Bereich (z.B. ElectriCChain Solar Project))
- Chain of Shipping (für Bereich Handel, Schifffahrt und Transport)

## **weitere Anwender / Nutzer / Global-Player / ...:**

- Ambisafe
- bitse
- Chronicled
- Consensys
- Distributed
- Hashed
- Ledger
- skuchain
-

---

#### **4.x.y.4. Anwendungsbereich: Energie-Sektor**

Übergang zu regenerativen Energie-Trägern / -Quellen

Probleme

viele Produzenten, kleine Produktions-Mengen; wenig planbar, geringe Stabilität während der Einspeisung, ...

? Abrechnung; stabile Strom-Versorgung

Slock.it in Zusammenarbeit mit RWE

Energie-Versorgung von / an Auto-Akku-Ladestationen (Electric Vehicle Charging)

Chain of Solar

Erfassung von Produktion und Verbrauch von verschiedenen Nutzern → genauere Planung und Abrechnungen

Forschungs-Projekt

Hersteller, Kunden / Nutzer, Forschung

LO3ENERGY

Lokale (New York, Brooklyn) Produktion von Solarstrom auf dem Dach, Energie-Speicherung in Akku's, Eigen-Verbrauch und Bereitstellung der Überschüsse für Nachbarn (Brooklyn Microgrid Project)

(Transacting Local Energy with Neighbors)



---

## **4.x.y.5. Anwendungs-Bereich: Logistik**

Logistik ist geprägt von komplexen Netz aus:

- Produzenten
- Vertrieb
- Händlern
- Beschaffern
- Lageristen
- Verkäufern
- Nutzern

(Spplier, Producer, Distributor, 3PL, Retailer, Store, Customer)

hohe Dezentralität und unterschiedlich skalierte Leistungs-Träger und -Nutzer

Lieferketten (Nachvollziehbarkeit von Ursprüngen)

Herkunfts-Nachweise / Originalität

viele Daten in unterschiedlichen Formaten und Währungen / Abrechnungs-Einheiten

Teilnehmer haben sehr unterschiedliche Berechtigungen Daten in der Kette zu verändern / neu hinzuzufügen oder zu löschen

Supply Chain Management (Lieferketten-Management)

unklare Vertrauens-Verhältnisse werden über Blockchain-Techniken stabilisiert

Bezahlung und Protokollierung

Ausfallsicherheit

Einbeziehung von Smart Contracts und Oracle's als Basis-Technologie

Einbindung von IoT (z.B. RFID od. Sensoren, ..., künstliche DNA)

Einhaltung von Transport-Bedingungen; Herkunfts-Nachweis

### **modum**

Handel und Transport von Medikamenten, Einbindung von IoT in die Lieferkette

### **IBM und Maersk**

Lösung für Schifffahrts- und Logistik-Industrie

### **Foxconn und Dianrong**

Blockchain-basierte Lieferketten-Finanz-Plattform

derzeit in Bereichen Automobil-, Elektronik- und Bekleidungs-Industrie

Verbesserung der Einfachheit, Transparenz und Überschaubarkeit von Transaktionen (Zahlungen und Warenbewegungen)

Verbesserung der Authentifizierung der (wechselnden) Partner

insgesamt Kosten-Ersparnis anvisiert

---

## **4.x.y.6. Anwendungsbereich: Identitäts-Management**

Problem bei der Identität / Identitäts-Prüfung / Account-Management  
im Internet  
für jeden Web-Dienst / ... eigene Identität notwendig

wenige Konsortien bieten Identitäts-Übernahmen (google, facebook, twitter, Xing, Linkin, ...) an  
Datenschutz aber unklar

### **Zookos Dreieck:**

Trilemma, da in (einem) Netzwerken gleichzeitig nur zwei von eigentlich drei angestrebten Eigenschaften erfüllt / optimiert sein können:

- Dezentralität (Verzicht auf eine vertrauens-würdige, zentrale Instanz)
- Sicherheit / Authentizität ((garantierte) kryptographische Sicherheit mit nur einem Schlüsselpaar)
- Aussagekräftigkeit (für Menschen lesbare Namen (und nicht automatisch generierte Zeichenfolgen))

durch Blockchain soll es möglich sein alle drei Eigenschaften (optimal) zu realisieren  
Nutzer gibt den Datenbereich frei, der für seinen Service-Anbieter notwendig ist (beim Arzt → Gesundheits-Daten, ...) → Self Sovereign Identity

Nutzer stellt (über seinen User Agent) Behauptungen im System auf (Ersteller)  
das können z.B. die Adresse, Ausweise, Führerschein und die Bank-Verbindung sein  
andere Nutzer des Systems bestätigen nun die Behauptungen (Verifizierer: Bank-Verbindung wird über die Bank verifiziert usw. usf.)  
Daten-Speicherung in Blockchain (Behauptungs-Register, ID-Register)

### **FRAUNHOFER-Projekt: Volksverschlüsselung.de**

in Zusammenarbeit mit Telekom  
<https://volksverschlueselung.sit.fraunhofer.de/>

### **Blockstack**

Vorreiter; erster Anwender / Anbieter  
→ <https://blockstack.org>  
Bereitstellung von: Identity, Storage, Tokens  
DNS-Daten  
plant Blockchain-Internet

in der Blockchain werden nur Änderungen verarbeitet → Blockchain Layer  
Virtualchain Layer → eigentliche System-Intelligenz (Erzeugung von Hash-Werten und Verknüpfung der verschiedenen Daten-Bereiche)  
Blockchain- und Virtualchain-Layer gehören zur Steuerungs-Ebene  
Routing- und Storage-Layer bilden Daten-Ebene  
Routing Layer (Verteilung der Daten)  
Storage Layer sind dann die Anwendungen / Dienste im System (Cloud-Speicher des Systems)  
z.B. amazonS3, DropBox, Microsoft Azure, Personal Drive, BitTorrent, ...

Bezahlung / Verrechnung über Krypto-Währungen

Anmeldung / Nutzung des Systems: (erzeugen einer Blockstack ID)

---

→ <https://blockstack.org/install>  
auch Beteiligung als Full Node möglich

**uport**

**sovrin (identity for all)**

**Jolocom (Linking outside the box)**  
deutsch

**Blockchain Helix**  
deutsch

---

## 4.x.y. Blockchain selber programmieren

### **Links:**

<https://tools4noobs.com> (Prüfziffern / Hash)

<https://andersbrownworth.com/blockchain/> (Visualisierung Blockchain)

### **4.x.y.0. Grundlagen / Ausgangspunkt**

einfache Liste von Transaktionen

### **4.x.y.1. einfache Transaktions-Historie**

verkettete Liste

### **4.x.y.2. Manipulations-Möglichkeiten der Transaktions-Historie**

Angriff auf letzte Transaktion  
Fälschung von Ziel und Betrag

Angriff auf Historie  
Fälschung von Ziel und Betrag

### **4.x.y.3. Absicherung der Transaktions-Historie**

Prüfsummen

Hash-Werte

work of prof

---

work of

## 4.x.y.4. ein Blockchain-System a'la Bitchoin

### 4.x.y.4.1. ein Blockchain-System a'la Bitchoin in Python

funktionierende Implementierung von Ivo STILLER  
Q: Weiterbildung IQSH "" 22.03.2022

start.py (Haupt-Programm)

```
1 import Block
2 import Blockchain
3
4 B = Blockchain.BlockChain()
5 B1 = Block.Block()

B1.addCoinBase("Soeren",10)
B1.addCoinBase("Laura",5)
B1.addCoinBase("Cedric",7)
B1.addToken("Soeren", 2, "Ivo")
B1.addToken("Soeren", 4, "Laura")
B1.addToken("Cedric", 7, "Stefan")

B.addBlock(B1)

B2 = Block.Block()
B2.setBlocknummer(2)
B2.addCoinBase("Daniel",20)
B2.addToken("Daniel", 20, "Finia")
B2.addToken("Soeren", 2, "Finia")

B.addBlock(B2)

B.giveChain()
B.verifyChain()
```

Block.py (Klasse: Block)

```
1 import Coinbase
2 import Token
3 import hashlib
4
5 class Block :
    def __init__(self) :
        self.blockNummer = 0
        self.nonce = 0
        self.inhalt = ""
        self.hash = ""
        self.prev = ""
        self.nextBlock = None
        self.coinBases = []
        self.Tokens = []

    def getCoinbases(self) :
        return self.coinBases

    def addCoinBase(self, r, v) :
        self.coinBases.append(Coinbase.CoinBase(r,v))
```

```

def addCoinBase(self, r, v) :
    self.coinBases.append(Coinbase.CoinBase(r,v))

def getTokens(self) :
    return self.Tokens

def addToken(self, s, v, r) :
    self.Tokens.append(Token.Token(s, v, r))

def getNextBlock(self) :
    return self.nextBlock
def setNextBlock(self, nextBlock) :
    self.nextBlock = nextBlock

def getPrev(self) :
    return self.prev
def setPrev(self, prev) :
    self.prev = prev

def getBlocknummer(self) :
    return self.blockNummer
def getNonce(self) :
    return self.nonce

def getHash(self) :
    self.hash = self.SHA256(str(self.getBlocknummer() +
self.getNonce()) + self.getInhalt() + self.getPrev())
    return self.hash

def getInhalt(self) :
    coinbase = "";
    for c in self.coinBases :
        coinbase = coinbase + c.toString()

    token = "";
    for t in self.Tokens :
        token = token + t.toString()

    inhalt = coinbase + token
    return inhalt;

def setBlocknummer(self, blockNummer) :
    self.blockNummer = blockNummer

def setNonce(self, nonce) :
    self.nonce = nonce

def SHA256(self, text) :
    return hashlib.sha256(text.encode("utf-
8")).hexdigest()

def MineBlock(self) :
    self.nonce = -1
    while (not self.hash.startswith("0000")) :
        self.nonce = self.nonce + 1
        t = str(self.getBlocknummer() +
self.getNonce()) + self.getInhalt() + self.getPrev()
        self.hash = self.SHA256(t)

```

Coinbase.py (Klasse: Coinbase ())

```

1 class Coinbase :
2
3     def __init__(self, r, v) :
4         self.Recipient = r
5

```

```

        self.Value = v

    def getRecipient(self) :
        return self.Recipient
    def setRecipient(self, Recipient) :
        self.Recipient = Recipient

    def getValue(self) :
        return self.Value
    def setValue(self, Value) :
        self.Value = Value

    def toString(self) :
        return self.Recipient + " ; " + str(self.Value) + "
; "

```

### Blockchain.py (Klasse Blockchain (Blockkette))

```

1 import Coinbase
2 import Token
3 import Block
4
5 class Blockchain :
    firstBlock = None

    def addBlock(self, B) :
        if(self.firstBlock is None) :
            B.setBlocknummer(1)
            B.setPrev("0")
            B.MineBlock()
            self.firstBlock = B
        else:
            Iterator = self.firstBlock
            while(not Iterator.getNextBlock() is None) :
                Iterator = Iterator.getNextBlock()

            B.setBlocknummer(Iterator.getBlocknummer()+1)
            B.setPrev(Iterator.getHash())
            B.MineBlock()
            Iterator.setNextBlock(B)

    def giveChain(self) :
        Iterator = self.firstBlock
        while(not Iterator is None) :
            print("Block " +
str(Iterator.getBlocknummer()))
            print("Inhalt: " + Iterator.getInhalt())
            print("Hash: " + Iterator.getHash())
            print("Nonce: " + str(Iterator.getNonce()) +
"\n")
            Iterator = Iterator.getNextBlock();

    def verifyChain(self) :
        Iterator = self.firstBlock
        while(not Iterator is None) :
            if(Iterator.getHash().startswith("0000")) :
                Iterator = Iterator.getNextBlock();

            else:
                print("Chain invalid at Block " +
str(Iterator.getBlocknummer()) + "!!!")
                return
            print("Chain valid!")

```

### Token.py (Klasse: Token (Knoten, Eintrag))

```

1 class Token :
2
3     def __init__(self, s, v, r) :
4         self.Sender = s
5         self.Value = v
6         self.Recipient = r
7
8     def getRecipient(self) :
9         return self.Recipient
10    def setRecipient(self, Recipient) :
11        self.Recipient = Recipient
12
13    def getValue(self) :
14        return self.Value
15    def setValue(self, Value) :
16        self.Value = Value
17
18    def getSender(self) :
19        return self.Sender
20    def setSender(self, Sender) :
21        self.Sender = Sender
22
23    def toString(self) :
24        return self.Sender + " ; " + str(self.Value) + " ; "
25        + self.Recipient + " ; "

```

#### 4.x.y.4.2. ein Blockchain-System a'la Bitchoin in JAVA

funktionierende Implementierung von Ivo STILLER

Q: online-Weiterbildung IQSH "" 22.03.2022

Start.java (Haupt-Programm)

```

1 package Java;
2
3 import java.security.NoSuchAlgorithmException;
4
5 public class Start {
6
7     public static void main(String[] args) throws
8     NoSuchAlgorithmException {
9
10        Blockchain B = new Blockchain();
11
12        Block B1 = new Block();
13        B1.addCoinBase("Soeren",10);
14        B1.addCoinBase("Laura",5);
15        B1.addCoinBase("Cedric",7);
16        B1.addToken("Soeren", 2, "Ivo");
17        B1.addToken("Soeren", 4, "Laura");
18        B1.addToken("Cedric", 7, "Stefan");
19        B.addBlock(B1);
20
21        Block B2 = new Block();
22        B2.addCoinBase("Daniel",20);
23        B2.addToken("Daniel", 20, "Finia");
24        B2.addToken("Soeren", 2, "Finia");
25        B.addBlock(B2);
26
27        B.giveChain();
28    }
29 }

```



```
        B.verifyChain();
    }
}
```

### Block.java (Klasse: Block)

```
1 package Java;
2 import java.nio.charset.StandardCharsets;
3 import java.security.MessageDigest;
4 import java.security.NoSuchAlgorithmException;
5 import java.util.ArrayList;

public class Block {
    private long blockNummer = 0;
    private long nonce = 0;
    private String inhalt = "";
    private String hash = "";
    private String prev = null;
    private Block nextBlock = null;
    private ArrayList<CoinBase> CB = new ArrayList<>();
    private ArrayList<Token> T = new ArrayList<>();

    public Block(){

    }

    public ArrayList<CoinBase> getCoinbases() {
        return CB;
    }

    public void addCoinBase(String r, double v) {
        CB.add(new CoinBase(r, v));
    }

    public ArrayList<Token> getTokens() {
        return T;
    }

    public void addToken(String s, double v, String r) {
        T.add(new Token(s, v, r));
    }

    public Block getNextBlock() {
        return nextBlock;
    }

    public void setNextBlock(Block nextBlock) {
        this.nextBlock = nextBlock;
    }

    public String getPrev() {
        return prev;
    }

    public void setPrev(String prev) {
        this.prev = prev;
    }

    public long getBlocknummer() {
        return blockNummer;
    }

    public long getNonce() {
        return nonce;
    }
}
```

```

    public String getHash() throws NoSuchAlgorithmException
    {
        hash = SHA256(getBlocknummer() + getNonce() +
getInhalt() + getPrev());
        return hash;
    }

    public String getInhalt() {
        String coinbase = "";
        for (CoinBase c : CB) {
            coinbase = coinbase + c.toString();
        }
        String token = "";
        for (Token t : T) {
            token = token + t.toString();
        }
        inhalt = coinbase + token;
        return inhalt;
    }

    public void setBlocknummer(long blockNummer) {
        this.blockNummer = blockNummer;
    }

    public void setNonce(long nonce) {
        this.nonce = nonce;
    }

    public String SHA256(String text) throws
NoSuchAlgorithmException {
        MessageDigest digest =
MessageDigest.getInstance("SHA-256");
        byte[] bytes = di-
gest.digest(text.getBytes(StandardCharsets.UTF_8));
        StringBuffer result = new StringBuffer();
        for (byte byt : bytes) {
            result.append(Integer.toString((byt & 0xff) +
0x100, 16).substring(1));
        }
        return result.toString();
    }

    public void MineBlock() throws NoSuchAlgorithmException
    {
        nonce = -1;
        while (!hash.startsWith("0000")) {
            nonce = nonce + 1;
            String t = getBlocknummer() + getNonce() +
getInhalt() + getPrev();
            hash = SHA256(t);
        }
    }
}

```

### CoinBase.java (Klasse: CoinBase)

```

1 package Java;
2
3 public class CoinBase {
4     private String Recipient = "";
5     private double Value = 0;

    public CoinBase(String r, double v){
        Recipient = r;
        Value = v;
    }
}

```

```

    }

    public String getRecipient() {
        return Recipient;
    }

    public void setRecipient(String Recipient) {
        this.Recipient = Recipient;
    }

    public double getValue() {
        return Value;
    }

    public void setValue(double Value) {
        this.Value = Value;
    }

    public String toString(){
        return Recipient + " ; " + Value + " ; ";
    }
}

```

### Blockchain.java (Klasse: Blockchain)

```

1 package Java;
2
3 import java.security.NoSuchAlgorithmException;
4
5 public class Blockchain {
    private Block firstBlock = null;
    public void addBlock(Block B) throws
    NoSuchAlgorithmException{
        if(firstBlock == null){
            B.setBlocknummer(1);
            B.setPrev("0");
            B.MineBlock();
            firstBlock = B;
        }
        else
        {
            Block Iterator = firstBlock;
            while(Iterator.getNextBlock()!=null)
            {
                Iterator = Iterator.getNextBlock();
            }

            B.setBlocknummer(Iterator.getBlocknummer()+1);
            B.setPrev(Iterator.getHash());
            B.MineBlock();
            Iterator.setNextBlock(B);
        }
    }
    public void giveChain() throws
    NoSuchAlgorithmException{

        Block Iterator = firstBlock;
        while(Iterator != null){
            System.out.println("Block " +
            Iterator.getBlocknummer());
            System.out.println("Inhalt: " +
            Iterator.getInhalt());
            System.out.println("Hash: " +
            Iterator.getHash());
            System.out.println("Nonce: " +
            Iterator.getNonce() + "\n");
        }
    }
}

```

```

        Iterator = Iterator.getNextBlock();
    }
}
public void verifyChain() throws
NoSuchAlgorithmException{
    Block Iterator = firstBlock;
    while(Iterator != null){
        if(Iterator.getHash().startsWith("0000")){
            Iterator = Iterator.getNextBlock();
        }
        else{
            System.out.println("Chain invalid at Block
" + Iterator.getBlocknummer() + "!!!");
            return;
        }
    }
    System.out.println("Chain valid!");
}
}
}

```

### Token.java (Klasse: Token)

```

1 package Java;
2
3 public class Token {
4     private String Recipient = "";
5     private String Sender = "";
    private double Value = 0;

    public Token(String s, double v, String r){
        Sender = s;
        Value = v;
        Recipient = r;
    }

    public String getRecipient() {
        return Recipient;
    }

    public void setRecipient(String Recipient) {
        this.Recipient = Recipient;
    }

    public double getValue() {
        return Value;
    }

    public void setValue(double Value) {
        this.Value = Value;
    }

    public String getSender() {
        return Sender;
    }

    public void setSender(String Sender) {
        this.Sender = Sender;
    }

    public String toString(){
        return Sender + " ; " + Value + " ; " + Recipient +
" ; ";
    }
}
}

```



---

## **5. Datenschutz und Datensicherheit**

---

## **6. Sicherheit in Datennetzen**

Authentifizierung

Vertraulichkeit

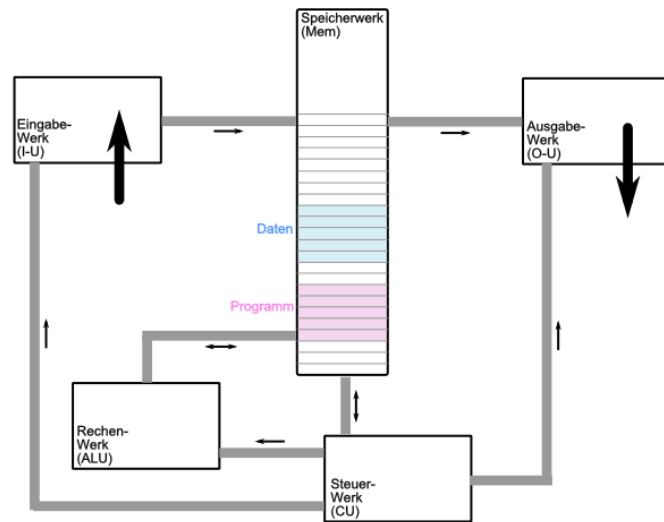
Signierung

Signaturen

Verschlüsselung / Kryptographie

komplexe Aufgaben (z.B. zur Vorbereitung auf eine Klausur)

- 1.
- 2.
3. Nebenstehende Skizze von einem Rechner-Konzept tauchte vor Kurzem auf. Es stammt aus der Zeit VON NEUMANN's. Ist dieses eines der verschollenen neuartigen Rechner-Architekturen nach denen die Geschichte der Informatik schon lange sucht? Beurteilen Sie das Modell!



- 4.



---

## **7. Netzwerk-Virtualisierung**

Q: wesentlich basierend auf OpenHPI-Kurs "Netzwerk Virtualisierung" W. BOEDDINGHAUS (04. – 25. September 2019)

### ***Situation vor Virtualisierung***

- pro Service eine Hardware (typischerweise ein Server)
- viel Hardware und hohe Kosten für wenig Leistung
  - aufwendige Anschaffung, Wartung, Administration, Betriebskosten, Platz / Räume
  - Geräte und Bestandteile oft sehr speziell (ev. optimiert)
- schlechte Ausnutzung der einzelnen Server (da keine kontinuierliche Benutzung)
- Probleme mit Ersatzteilen / Ausfällen
- 

### ***Virtualisierung bringt ...:***

- hohe Ausnutzung und Auslastung der Server, da viele Service's auf einer Hardware laufen
- weniger Hardware-Bedarf
  - universelle (billigere) Hardware
  - geringere Probleme bei der Beschaffung von Ersatzteilen
- geringe Betriebskosten
- 

### ***Netzwerk-Virtualisierung bringt ...***

- Entkopplung von Hard- und Software
- Entkopplung von Hardware und Diensten
- Datenfluss erfolgt nur noch teilweise auf der Ebene der Verkabelung
  - es gibt jetzt
    - logische Sicht (Datenfluss zwischen Service's)
    - physikalische Sicht (echte Verkabelung usw.)
- 

### ***Trennung der Netze in***

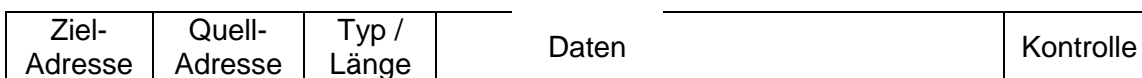
- **Underlay** echte Verkabelung / physikalische Schicht  
entspricht den traditionellen Netzwerk-Strukturen  
bestimmt die Gesamtstruktur und die Stabilität (Ausfälle, Störungen, Fehler wirken bis zum Overlay durch)  
Basis (Fundament eines Gebäudes)
  - Hardware (Kabel, Netzwerkkarten, Router, Switche, Rechner, ...)
- **Overlay** virtuelle Schicht / logisches Netzwerk  
aufgesetzte Schicht (auf Underlay) mit eigenen Strukturen und Datenflüssen  
Aufbauten (Stockwerke eines Gebäudes)
  - nur virtuelle Geräte (virtRouter, virtSwitche, ...)
  - Datenkann keine (bzw. nur kleine) Fehler im Underlay überdecken

Probleme sind meist schwieriger zu bearbeiten, da sowohl das Underlay und das Overlay betroffen sind oder sein können  
 Tunnel im Overlay bringen weitere Probleme, da nun 3 Ebenen zu betrachten sind

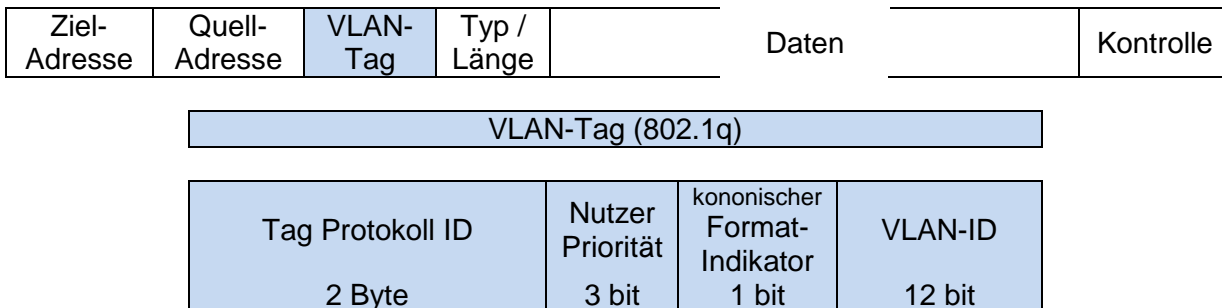
## 7.1. virtuelle LAN's - VLAN

relativ alte Technik, häufig genutzt  
 verwendet bei der Unterteilung der LAN-Infrastruktur  
 passiert im OSI-Modell auf Layer 2 (Ebene der Switches)

klassisches Ethernet-Paket (Ethernet-Frame)



Ethernet-Paket mit VLAN-Tag



es gibt eine Stelle in Netzwerk, an der der VLAN-Tag eingefügt (angebracht) wird sowie eine Stelle, wo er wieder entfernt (gelöscht) wird.

Zwischen-Geräte müssen den VLAN-Tag transportieren können (nun ja verlängertes Paket (um 4 Byte))

in Netzwerken muss beachtet werden das die MTU (Maximum Transport Unit) entsprechend angepasst wird

für die Endgeräte bleibt es aber bei den üblichen 1500 Byte pro Ethernet-Paket (plus ev. Header)

für das eigentliche Endgerät (z.B. PC) ist das virtuelle Netzwerk nicht sichtbar (ist ja an das physikalische Netzwerk angeschlossen)

zwischen den Switches von virtuellen LAN's befinden sich Trunk's. Über sie laufen die VLAN-Pakete

Trunk ist Begriff aus der CISCO-Welt (ev. Tag-Port's, Tag-Leitung)

die Access-Ports sind dann die Stellen, wo die VLAN-Tag's eingefügt bzw. gelöscht werden

über die Nutzer-Priorität werden Quality of Service und Class of Service festgelegt, um Vorränge zu realisieren

soll in einem normalen Ethernet eine Priorisierung erfolgen, dann muss der Weg über VLAN-Tag's gemacht werden. Das normale Ethernet auf Layer 2 lässt keine Priorisierung zu!.

kanonischer Format-Indikator stammt aus der Token-Ring-Welt (bei Ethernet auf 0 gesetzt) heute ersetzt durch "Drop eligible"-Indikator (DEI), der festlegt, ob Paket bei Netzwerk-Überlastung gedropt (übergangen) werden kann (ähnlich IP-Paketen)

VLAN-ID ist Wert größer als 0x000 und kleiner als 0xFFF, damit sind praktisch 4094 (4096-2) VLAN's möglich  
in extended VLAN längere VLAN-ID möglich

#### QinQ

doppelte Anbringung von VLAN-Tags im VLAN-Paket (nochmal 4 Byte länger)  
erster Tag für die "Betriebs-interne Virtualisierung) und der zweite Tag für die Übertragung zwischen Netzwerken  
damit können auch Service-Provider in ihren LAN's virtualisieren und VLAN-Pakete (aus anderen VLAN's) transportieren

Ziel-Adresse	Quell-Adresse	VLAN-Tag	VLAN-Tag	Typ / Länge	Daten	Kontrolle
--------------	---------------	----------	----------	-------------	-------	-----------

## 7.2. Port-Channels

	VLAN	Port-Channel
	teilt Netzwerke auf Teilen der Bandbreite Trennen der Daten / Pakete durch Markierungen	bündeln physikalische Interface's Zusammenfassung der Bandbreite (Erhöhung) Redundanzen ausnutzen

gebraucht werden mindestens 2 Leitungen, die auch immer die gleiche Bandbreite haben müssen

der Typ darf sich unterscheiden. Kupfer- und Glasfaser lassen sich also kombinieren  
typisch bis zu 8 Leitungen, nicht aktive Leitungen (z.B. 9. od. 10 Leitung) sind Reserve  
praktisch immer nur maximal 8 aktiv

jeder Zugriff (z.B. Datenbank-Abfrage, Web-Seiten-Aufruf, Ping, ...) läuft für sich über einen Channel

bei einzelnen Nutzungen ist Ungleich-Verteilung normal, bei steigender Nutzer-Zahl verteilen sich die Band-Auslastungen praktisch gleichmäßig über alle Channel's

unterschiedliche Verteilungen möglich, nach:

- Absender-IP
- Ziel-IP
- Absender-Port
- Ziel-Port
- MAC-Adressen

optimal sind 2, 4 oder 8 Leitungen

---

### **Vorteile / Ziele der Port-Channel-Technik**

- mehr Bandbreite zur Verfügung stellen
- Bandbreiten besser ausnutzen / auslasten
- höhere Ausfallsicherheit / Leitungs-Redundanz
- Austausch defekter Hardware im laufenden Betrieb (Hot-plug-In)
- Bereitstellung von Reserve-Leitungen
- Erzielen von übernormalen Daten-Durchsätzen
- 

### **übliche Port-Channel-Protokolle**

- **Link Aggregation Control Protocol)  
LACP** IEEE  
mehr verbreitet  
auch für ungünstige Leitungs-Anzahlen  
geeignet
- **Port Aggregation Protocol  
PAGP** Cisco

für die Konfiguration am Endgerät ändert sich nichts  
alle erweiterte Konfiguration wird über logische Konfiguration der Pot-Channel-Interface's  
realisiert

das ist dann Layer 3 (OSI)

Konfiguration von IP-Adressen, QoS, IPsec

auf Layer 2 arbeiten die Trunk's

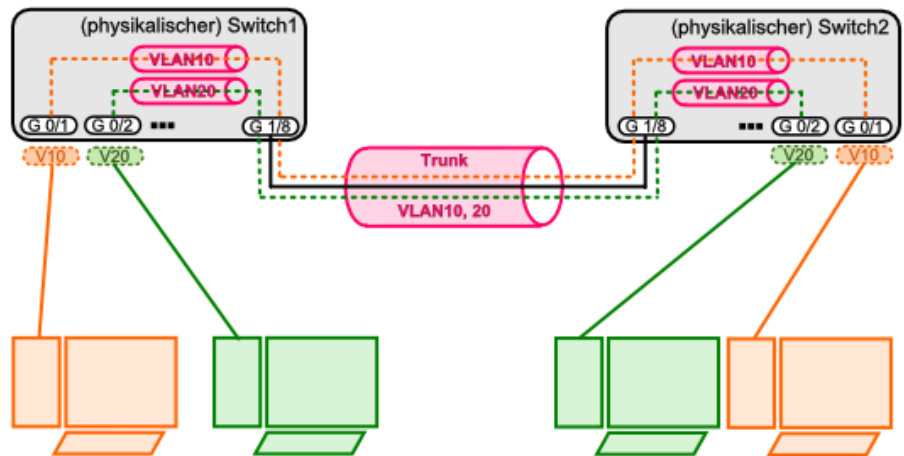
bei MLAG (Multi-Channel-Link-Aggregation) wird ein Server mit 2 Netzwerkkarten ausgestattet, diese werden als ein Port-Channel konfiguriert → doppelte Bandbreite  
die Leitungen gehen zu zwei Switchen, je Switch also eine Leitung  
beide Switche sind dann noch einmal untereinander verbunden  
fällt eine Leitung aus, dann kann trotzdem das gesamte Netz versorgt werden (allerdings nur noch mit der einfachen Bandbreite)  
z.B. in Cloud-Systemen genutzt

## **7.3. Tunnel**

Unterscheidung / Fragen

- Was soll transportiert werden? Mit welchem Protokoll (Gast-Protokoll) sollen die Original-Daten transportiert werden?
- Welche IP-Adressierung wird gewählt?
- Welche Netzwerk-Technik wird verwendet (Token-Ring, Ethernet, ...)
- Gibt es VPN's?
- Soll die Übertragung verschlüsselt erfolgen?
  
- Welches Protokoll soll zum Transport (Transport-Protokoll) genutzt werden?
- Welche IP-Adressierung wird gewählt?

- Welche Netzwerk-Technik wird verwendet (Token-Ring, Ethernet, ...)



Transport-Protokoll → Underlay

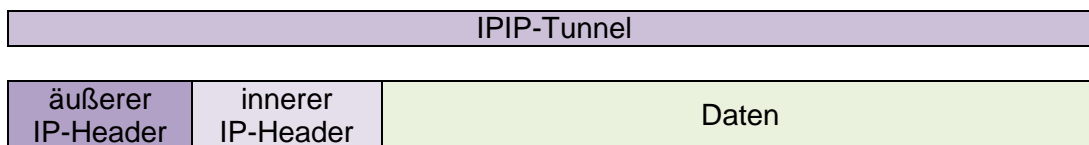
Gast-Protokoll → Overlay

→ zwei (IP-)Header hintereinander, was wieder die Paketgröße ändert und bei der Konfiguration der MTU beachtet werden muss

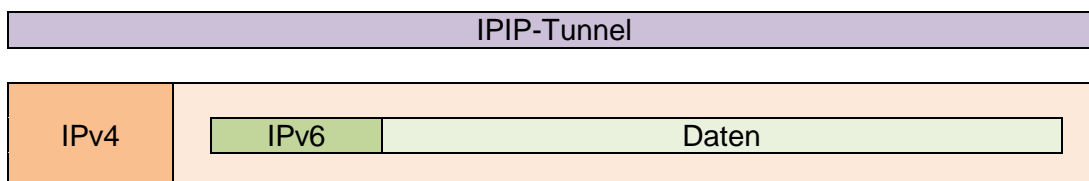
der Tunnel ist eine logische Verbindung (unsichtbar auf der Ebene des Overlay) über dem Underlay  
ein TraceRoute zeigt nichts vom Underlay

wenn es gewünscht wird, dann können Informationen zum QoS und TTL (Time to Life) auch vom inneren auf den äußeren Header übertragen werden

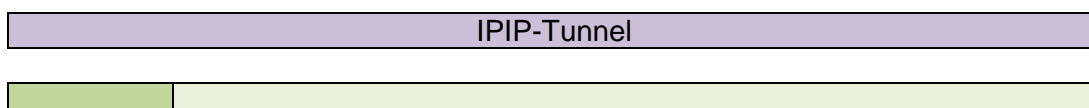
Hinzufügen und Entfernen des zusätzlichen Header an den Tunnel-Enden

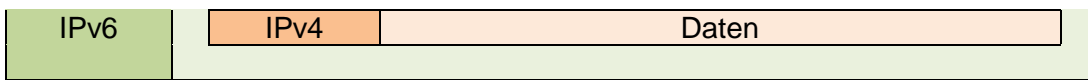


über IPIP-Tunnel lassen sich nur IP-Pakete transportieren  
es gibt kein Spanning Tree () und kein IS-IS-Routing ( auf Layer 2)



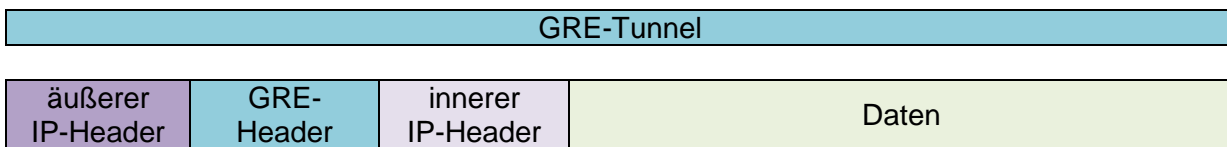
oder auch:





### GRE-Tunnel

(generic route encapsulation)  
 transportiert alle Paket-Arten  
 verhält sich wie ein Ethernet-Kabel  
 Paket nochmals länger (als IPIP-Pakete)



Konfiguration unter Linux:

```
ip tunnel Interfacename mode ipip remote 1.1.1.1 local 2.2.2.2 ttl 255
                                     (Ziel-Adresse)   (lokale Adresse)
```

auf der anderen Seite des Tunnels:

```
ip tunnel Interfacename mode ipip remote 2.2.2.2 local 1.1.1.1 ttl 255
```

für GRE:

```
ip tunnel add Interfacename mode gre remote 1.1.1.1 local 2.2.2.2 ttl 255
```

## 7.4. Virtuelle private Netzwerke

Virtual Private Network (VPN)

virtuelles Netzwerk

meist zur sicheren Verbindungen von Firmen und / oder Mitarbeitern genutzt  
 aber auch zur Abhör-sicheren Verbindung zwischen Privat-Nutzern

### Arten von VPN's

- Dial in VPN**

temporäre Verbindung / Einwahl-Verbindung von Mitarbeitern bzw. Firmen-Außenstellen beim Rechenzentrum (Bedarfsleitungen)  
 im Rechenzentrum wird das VPN zentral verwaltet  
 auf dem entfernten Rechner ist ein VPN-Client  
 z.B.: Verbindung vom HomeOffice, Kunden-Standort, Flughafen, Cafe, Restaurant, Hotel, ... → Nutzer / Client ist mobil  
 nutzt als Underlay das Internet → keine Kontrolle über Datenströme  
 wechselnde IP-Adressen  
 schwangende Bandbreite (WLAN, LTE, ...) / Qualität der Leitung
- Site to Site VPN**

Vernetzung zwischen 2 Standorten (2 Firmen-Standorte, Rechen-

zentrum und Homeoffice des Mitarbeiters, ...)  
dauerhafte Verbindung  
oft über die Firewall konfiguriert  
nutzt als Underlay das Internet → keine Kontrolle über Datenströme  
Best Effort (minimalistische Dienstgüte-Zusicherung in Kommunikationsnetzen  
→ keine Garantie für Übertragung von Daten-Paketen)  
Abhängig von verfügbarer Bandbreite zwischen Standorten  
feste IP-Adressen

muss nicht verschlüsselt sein, ist es aber meistens  
an öffentlichen Orten / Netzwerken ist immer eine Verschlüsselung zu empfehlen  
mit IPSec

- für IPv4 und IPv6 geeignet
- praktisch alle Betriebssysteme unterstützen diese verschlüsselung
- sehr häufig verwendet
- Produkte:
  - Cisco ASA oder Router
  - Checkpoint Firewall
  - Fortinet Firewall
  - praktisch alle besseren Firewall's
- 

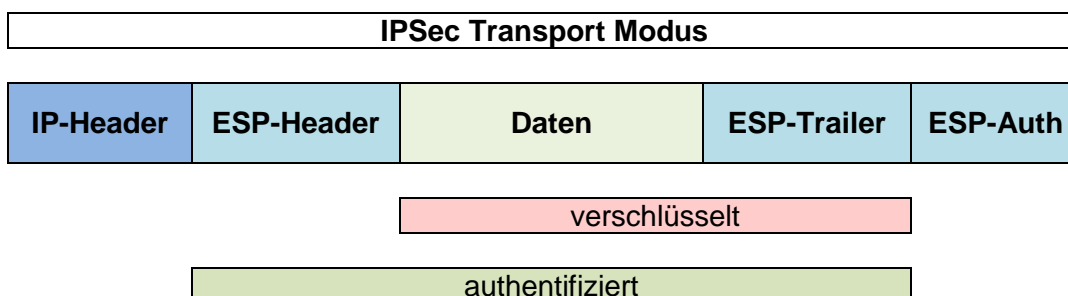
### OpenVPN

- mit eigener Verschlüsselung
- für IPv4 und IPv6 geeignet
- ist Open Source
- kostenfrei
- Produkte:
  - OpenVPN
  - Tinc VPN
  - Soft Ether (ev. problematisches Produkt?!)
- 

## IPSec

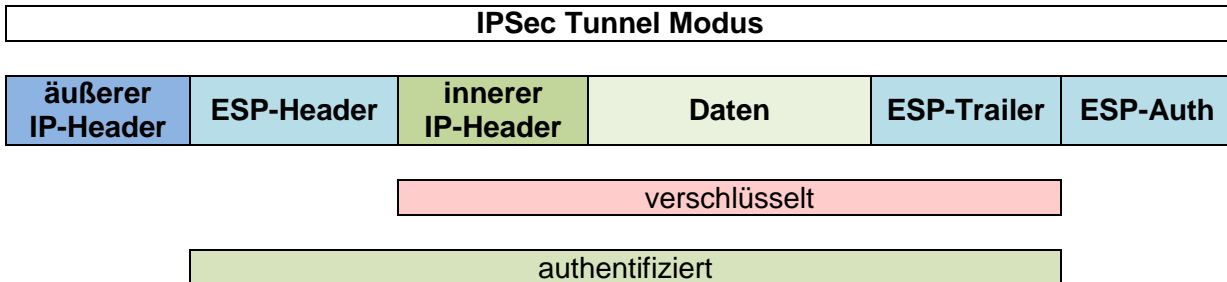
### Transport-Mode

Verschlüsselung des laufenden Datenstroms  
nur die Daten in den Paketen werden verschlüsselt, nicht die Header  
keine Konfiguration von Interface's  
das Routing erfolgt ganz normal über das Internet  
ESP .. encrypted security payload



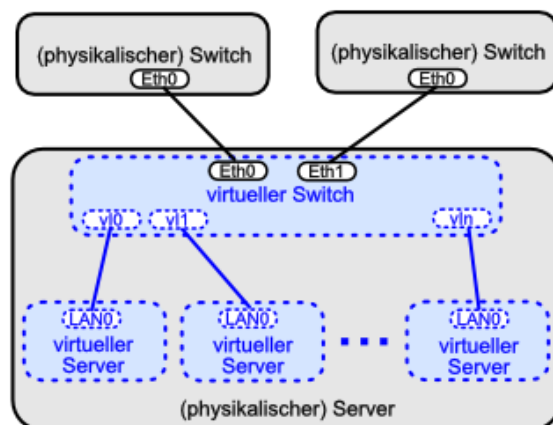
## Tunnel-Mode

es wird ein Interface erzeugt (Virtual Tunnel Interface (VTI))  
Datenpaket des Nutzers wird vollständig verschlüsselt  
es kommt ein neuer Header für den Tunnel hinzu  
Pakete verlängert



## 7.5. Linux virtueller Switch

virtuelle Switche benötigen immer ein reales, physikalisches Netz als Basis



virtuelle Switche z.B. in:

- Linux KVM
- Openstack
- VMWare
- AWS – Amazon Cloud
- Google Cloud
- Microsoft Azure
- Virtualbox

praktisch gleich, etwas unterschiedlich zu konfigurieren werden vom Server erwartet

### **Linux-basierte virtuelle Switche**

- **Linux Bridge** einfach; mehrere Bridge's einrichtbar  
verhält sich, wie normaler Switch  
eine Bridge kann ein physikalisches Interface enthalten (muss es aber nicht); dieses hat meist keine eigene IP-Adresse



---

mehr (Server nur noch über die IP der Bridge erreichbar)  
Interface's lernen MAC-Adressen  
Spanning Tree kann benutzt werden  
kann auch eine IP-Adresse bekommen (Layer 3)  
kann IPv4 und IPv6  
IPtables möglich und aus Sicherheitsgründen auch notwendig  
kann VLAN-Tags transportieren  
VLAN Interface's (eth0.100) können eingebunden werden  
es können auch für jedes VLAN eigene virtuelle Bridge's eingebaut werden

- **Open Virtual Switch OVS**

sehr komplex, Leistungs-fähiger als Linux Bridge  
Alternative zur Linux Bridge  
(sollte nicht gleichzeitig mit Linux Bridge verwendet werden → unübersichtlich)

OVS kann:

- VLAN tagging
- Portchannel / LACP
- Spanning Tree
- QOS
- Tunnel-Protokolle (GRE, VXLAN, LISP, IPSec)
- SPAN, RSPAN (Duplizieren des Daten-Verkehrs, z.B. zur Dokumentation und für Diagnosen (z.B. mit Wireshark))

Grundlage für Linux-Cloud (z.B. Openstack mit OVS)

- 

Kommando's zur Steuerung der Bridge

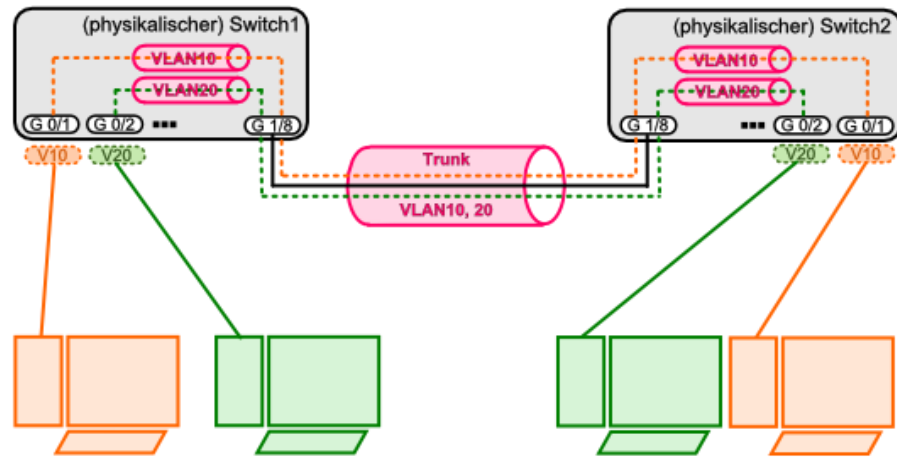
<b>brctl</b> ...	Bridge Control
addbr	erzeugen
delbr	löschen
addif	Interface hinzufügen
delif	Interface löschen
show	alle Bridge's anzeigen
showmacs	MAC-Adressen an der Bridge anzeigen

Kommando's zur Steuerung der Open Virtual Switch

<b>ovs-vsctl</b> add-br <i>Brigdename</i>	Erzeugen einer neuen Bridge mit dem Namen " <i>Brigdename</i> "
<b>ovs-vsctl</b> add-port <i>Brigdename Interface</i>	Erzeugen eines neuen Interface an der Bridge

Im Unterschied zur physikalischen Switch'es (Hardware-Switches) kann die Zuordnung der Interface's frei bestimmt werden. Man kann alle nutzen, muss es aber nicht.

```
ovs-vsctl add-port Meinebrigde MeinInterface tag= VLAN-Nummer
```



`ovs-vsctl add-bond Brigde Bond Interface1 Interface2 lacp=active`  
 Herstellen eines Interface-Bündels mit dem Namen Bond und den Mitgliedern Interface1 und Interface2

ein Bond kann aktiv/aktiv oder aktive/passiv arbeiten

### White Label Switch

nur Hardware wird gekauft  
 als Betriebssystem wird Linux genutzt  
 Ebtkopplung von Hardware und Software  
 volle Freiheit bei der Konfiguration

- Switching (OVS)
- Routing
- Container
- Python

führend ist hier Cumulus Linux, aber auch andere verfügbar

## 7.6. Linux als Router

Linux ist ein Server-Betriebssystem  
 statisches Routing ist immer mit dabei

aktuelle Routing-Tabelle über  
`ip route show`

es folgt typische Anzeige des Standard-Gateway's und des aktuellen Netzwerk inklusive des eigenen Interface's

für weitere Routing-Protokolle ist extra Software notwendig

---

## ausgewählte Routing-Protokolle

- **Quagga** typischer Routing Daemon aus dem Zebra-Projekt entstanden Quagga (Name eines ausgestorbenen Zebra's) ist ein Fork des Zebra-Projekt's als Paket installierbar kann:
  - RIP / RIPng
  - OSPFv2 / OSPFv3
  - BGP
  - IS-IS
- **Free Range Routing FFR** ist wiederum ein Fork von Quagga (durch google initiiert) heute sehr aktiv weiterentwickelt aktive Community kann mittlerweile zusätzlich (zu Quagga):
  - Openfabric (Weiterentw. von IS-IS)
  - LDP (MPLS)
  - EIGRP (ehem. Cisco, jetzt frei)
  - PIM
- **Bird** in Paaring-Point's
- 

Router teilen physikalische oder virtuelle Netzwerke in kleinere Layer-2- oder -3-Netzwerke aus Layer 2 noch stark störanfällig (z.B. Broadcast-Stürme)  
bei Layer 3 ist die Möglichkeit für Paket-Filterung oder der Einsatz einer Firewall gegeben  
Nachteil ist, dass ein Linux-Router immer auch ein vollständiges (Server-)Betriebssystem ist  
brauchen relativ viele Ressourcen  
haben immer auch anderweitige Angriffspunkte

für Cloud's werden viele kleine Netzwerke gebraucht – vornehmlich virtuelle  
hier wird virtuelles Routing also obligatorischer Bestandteil

## 7.6. VXLAN

Virtual extensible Local Area Network

Layer-2-Virtualisierung

Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks

während das VLAN nur 4'096 VNI's (Virtual Network Identifier) ermöglicht, sind es unter VXLAN 16 Mill.

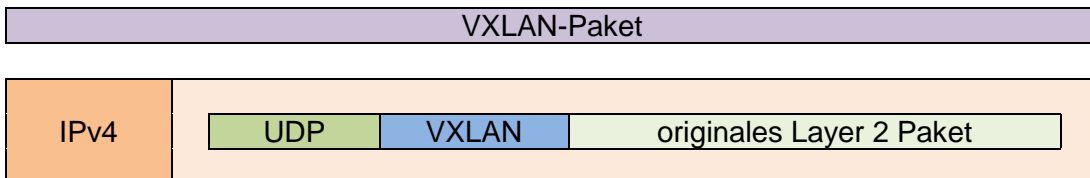
für Datacenter-Verbindungen (Datacenter Interconnect) → Verteilung von Daten auf mehrere Rechenzentren

das Verschieben von virtuellen Maschinen ist nun nicht mehr möglich, da getrennte Layer-2-Bereiche definiert sind

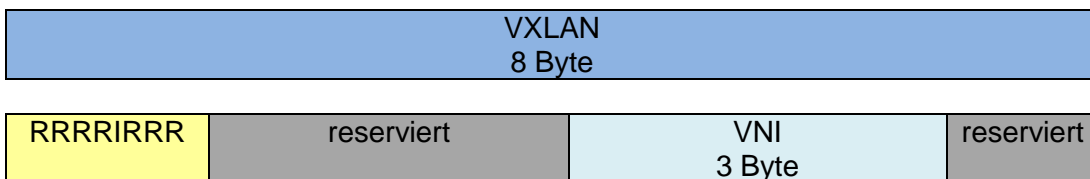
über VXLAN lassen sich zwei getrennte Layer-2-Bereichen untereinander verbinden

wenn Firewall's als Cluster laufen (je eine Firewall in jedem Rechenzentrum) → Verbindung der beiden Firewall's durch VXLAN

im VXLAN gibt es auch einen VTE (Virtual Tunnel Endpoint)  
 hier werden die originalen Pakete in die VXLAN-Pakete eingebaut / herausgelöst  
 lässt sich als Hard- oder Software realisieren  
 Software: z.B. in Vmware innerhalb eines Cluster's  
 Hardware: Kommunikation mit der Umwelt



MTU muss hier wieder ev. angepasst werden



	VLAN	VXLAN
<b>Unterschiede</b>	reines Layer 2 nur für geschwichte Umgebungen	Erweiterung von VLAN "routed" Layer 2 Pakete
	MAC-Adressen-Lernen über Flooding	MAC-Adressen-Lernen über Broadcast und BGP
<b>Gemeinsamkeiten</b>		

## 7.6. Virtual Routing and Forwarding - VRF

immer nur lokal auf dem speziellen Router  
 über eine Routing-Tabelle werden die Weiterleitung organisiert (Forwarding)  
 Pakete / Netzwerk-Verkehr wird über die Netzwerk-Adressen bestimmten Interface's zugeordnet

in der Start- bzw. Default-Tabelle gibt es nur eine Route  
 mittels VRF können beliebig viele und voneinander unabhängige Routing-Tabellen genutzt werden

verfügbar auf:

- Cisco
- Juniper
- Linux (neu)
- ...

durch eigenständige Routing-Tabellen ergeben sich vollständig virtualisierte Netzwerke so dass z.B. in jeder Tabelle mit dem gleichen Netzwerk (z.B. 10.0.0.8/8) gearbeitet werden kann

Netzwerke können auch überlappen (da sie sich ja gegenseitig nicht sehen / betreffen) (dies darf aber nicht in der zentralen Default-Routing-Tabelle passieren!)

jedem VRF sind Interface's zugeordnet, jedes Interface ist genau einem VRF zugeordnet

jedem Interface eines Router's wird eine Routing-Tabelle zugeordnet, dieses Interface kann auch virtuell oder Tunnel sein

## VRF mittels MPLS

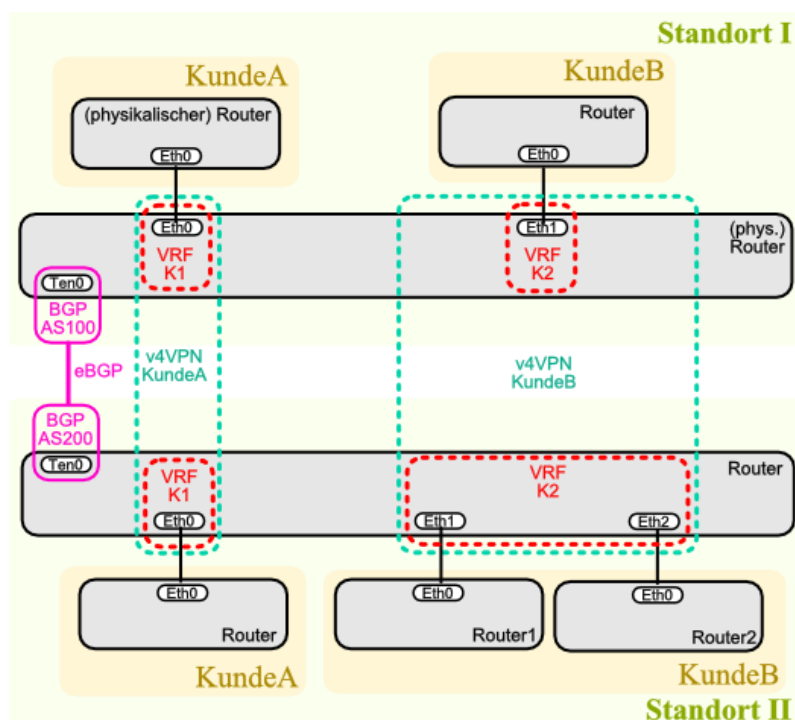
für die Vernetzung von Firmen-Standorten

2 virtuelle Netzwerke für Kunde 1 und 2 jeweils unabhängig voneinander

verbunden über physikalischer Router an verschiedenen Orten

für die Kunden sind es eigenständige – sichere- Netz-Verbindungen, die gemeinsam über eine physikalische Leitung gehen

für jeden Kunden ist sein Netzwerk vollständig transparent  
es gibt 3 Routing-Tabellen



eine Tabelle (die Standard-Routing-Tabelle) zum Austausch von Daten mit anderen (physikalischen) Routern (an anderen Orten)

für die Interface's der Kunden gibt es jeweils eine weitere Routing-Tabelle

Übergänge zwischen den virtuellen Netzwerken (Kunden-Netzwerke) ist nicht möglich (auch wenn beide z.B. ein gleiches Netzwerk nutzen)

## VRF zum Managen

es gibt eine weitere Routing-Tabelle für das Geräte-Management

dieses dient u.a. dem Schutz des Systems vor Angriffen

bleibt auch erreichbar, wenn Teile eines (Arbeits- / Nutz-)Netzwerkes nicht mehr funktionieren wird genutzt für:

- SSH

- Controller
- SNMP
- ...

bei vielen Routern ist Management-VRF standard-mäßig integriert  
eigenständiges (Management-)Netzwerk

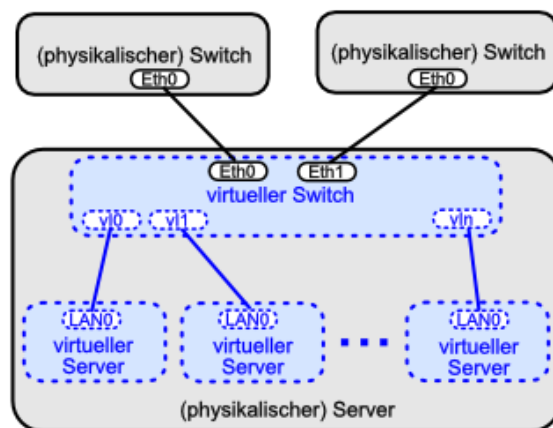
## 7.8. KVM (Kernel Virtual Machine)

fest in den Kernel von Linux eingebaut, kann zusätzlich als Modul geladen werden  
hierauf basieren die meisten Virtualisierungen unter Linux  
z.B. Openstack

Alternative ist: XEN  
Virtualisierung für Amazon Cloud und ähnliche Service's  
nutzt intern die Linux Bridge und oder OVS (Open V?? Switch)

virtuelle Teile gestrichelt

man nutzt Linux Bridge oder OVS (→ [7.5. Linux virtueller Switch](#))



auch die virtuellen Server brauchen ein Netzwerk zum Kommunizieren mit anderen Geräten usw. usf.

virtuelle Switche verbinden virtuelle Interface's mit physischen Interface's oder mit anderen virtuellen Interface's

Switch-Port zum Anschluss eines Server's bzw. eines Client's ist besonders wichtig  
nur hier ist der Daten-Verkehr eines Server's isoliert  
möglich sind u-a-:

- Access-Listen
- Security (z.B. RA Guard)
- QoS
- Mitlesen und Kopieren / Spiegeln von Daten (! ev. Problem mit den Datenschutz!)
- ...

## Tunnel

Punkt-zu-Punkt-Verbindungen  
auf Layer 2 (für's Bridging) oder 3 (mit IP-Nummer)  
z.B. genutzt für die Vernetzung von Cloud

---

ip tunnel Befehle für Linux  
OpenVPN ist mögliche Alternative

lassen sich verschlüsseln, z.B. mit:  
OpenVPN  
Wireguard  
TINC  
Softether

jeweils eigene Verschlüsselungs-Möglichkeiten und Optionen

Administration immer über die Shell möglich  
ip Command Suite ermöglicht Kontrolle aller Aspekte in einem Netzwerk  
mühselig und Fehler-anfällig

graphisches Konfigurations-Programm z.B. für OpenStack vorhanden

## 7.8. VMware

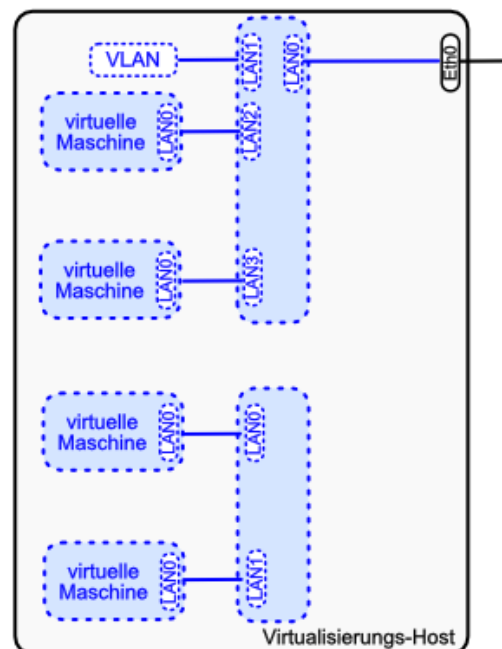
sehr Leistungs-fähige (professionelle)  
Virtualisierungs-Umgebung (Hypervisor)

in fast allen Unternehmen eingesetzt (entweder im eigenen Rechenzentrum oder auf entfernten Rechnern)  
man braucht aber ein Netzwerk

bietet virtuelle Switche und Device's

Beispiel-Netzwerk besteht aus drei Switchen  
Switch kann physikalisches Interface besitzen  
einem Switch können VLAN-Fähigkeiten zugeordnet werden

es gibt auch distributed Switches  
diese überspannen mehrere Hypervisor



braucht gemeinsamen Speicher

Virtuelle Maschinen lassen sich zwischen den beteiligten Servern (Hypervisoren) verschoben werden (Ressourcen müssen natürlich vorhanden sein)  
man erhält eine große, gemeinsame Layer-2-Zone (ev. problematisch) → Fehler und Vertrauen

solche Systeme sind relativ einfach zu planen und zu administrieren

Änderungen werden übertragen

einige Feature's sind nur unter den Distributed Switch zu erhalten:

- LACP
- Änderungen an der MTU

---

Einbindung von Router'n möglich  
Trennung erfolgt auf Layer 3  
man braucht mehr Ressourcen, da sie praktisch vollständige Betriebssysteme enthalten  
nutzung von Firewall's möglich (werden gerne auch als Router mißbraucht, sind aber keine)

## 7.8.x. VMware NSX

ist reine Cloud-Lösung im Datacenter; Enterprise-Lösung, nichts für zuhause  
durch wird VMware echte Netzwerk-Lösung, die völlig unabhängig vom Underlay ne Hardware  
verfügt über virtuelles Routing, Switching, Security sowie Loadbalancing  
Ziele sind:

- Schutz der Applikation (nicht des Servers)
- hoher Grad an Automatisierung
- Arbeiten mit Templates (Muster.Vorlagen)
- Anbieten von Self Service's () für Entwickler
- Reduktion von menschlichen Fehlern (z.B. Cut and Paste)
- weniger Arbeit auf der Kommandozeile (hoher Grad an Aufmerksamkeit notwendig)
- Reduktion der manuellen Arbeiten
- ...

### **früher / traditionell:**

Administrator beantragt Server, Verkabelung, erstellt Firewall-Regeln, beobachtet und konfiguriert das Loadbalancing

### **heute / modern:**

Administrator kann seine Ressourcen eigenständig managen  
das Datacenter stellt Rahmen-Bedingungen

Firewall ist zentrale Stelle als Point of Enforcement (hier können Regeln eingebaut werden)

sind z.B. mehrere SQL-Server eingebunden, dann kann nichts den Datenverkehr zwischen den beiden SQL-Servern verhindern (z.B. nach einem SQL-Angriff)  
keine Kontrolle des Traffic möglich

Policies kontrolliert den Datenverkehr → Switch-Ports

→ echte Microsegmentierung

Policies

- über Switches und Hardware wirksam
- ermöglichen freie Kombination von VM's
- können über mehrere VM's aus mehreren Netzwerken wirken
- trennen und führen zusammen
- Mitnahme der Policies z.B. bei Vmotion möglich



durch Microsegmentierung

- werden die VM's vollständig isoliert
- unabhängig vom Ort
- Policies wirken wie Firewall's
- VM's lassen sich zu Gruppen zusammenschließen

NSX bietet Distributed Logical Router

- liegt (vollständig) im Hypervisor-Kernel (→ schnell und Ressourcen-schonend)
- funktioniert über mehrere Hosts
- VM-Movement nimmt die Regeln mit
- bietet Control-Point im DLR (Distributed Logical Router)
- Peering nach außen
- OSPF, BGP, Static Routing
- ECMP
- Redistribute ist für Connected Routes möglich

Ost-West-Traffic (innerhalb des Rechenzentrums)

**Nord-Süd-Traffic (Kommunikation mit Außenwelt; User-Traffic) über Edge Device**

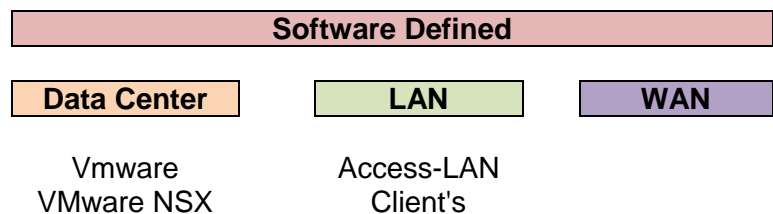
kann Routing (OSPF, BGP, Static), Firewall, VPN, Loadbalancer

sehr hoher Daten-Durchsatz

## 7.9. Software Defined WAN - SD-WAN

dient der Vernetzung von Firmen-Standorten mittels Virtuellen Netzwerk

Ziel ist Kontrolle über Internet-Breakout  
 schnelle Anbindung  
 hohe Flexibilität  
 schnelle Reaktion auf Leitungs-Ausfälle



	Microsegmentierung	SD-WAN
<b>Unterschiede</b>	für Client's und Datacenter  volle Kontrolle über das Underlay, weil es im Unternehmen liegt  Underlay kann SLA haben	Vernetzung von Standorten / Niederlassungen Steuerung des Weitverkehr Absicherung von Internet Breakout Internet als Underlay kein SLA im Internet schnelle Einbindung von Filialen / Niederlassungen / ...
<b>Gemeinsamkeiten</b>		

---

### Ziele von SD-WAN:

- Standorte mittels Internet-Provider vernetzen
- MPLS (Multi Protocol Label Switching) nutzen; passende Netzwerk dafür aufbauen (→ [7.6. Virtual Routing and Forwarding - VRF](#))

### Vorteile:

- gute Qualität
- z.B. gut geeignet für Voice, kleine Dateien, zeitkritischen Daten-Austausch (z.B. SAP)
- es lassen sich unterschiedliche Leitungs-Arten und –Qualitäten miteinander kombinierbar
- zentrale Verwaltung / Kontrolle des Gesamt-Netzwerkes
- ...

### Nachteile:

- teuer (deshalb üblicherweise knapp bemessen → Flaschenhals)
- oft kein Video, Übertragung großer Dateien schwierig
- lange Zeit Alternativlos
- Internet als Transportweg oft zu schlecht (Ausfälle, Support-Probleme (Wer ist am konkreten Problem Schuld?))
- kein Einfluss auf Entstörung
- zusätzlicher Planungs- und Test-Aufwand
- kein SLA möglich (läßt das benutzte Internet nicht zu)
- bei Leitungs-Ausfällen kann es zur Blockierung der App usw. kommen
- ...

heute ist Bandbreite kein Thema mehr, praktisch immer realisierbar, deshalb ist das Internet als Underlay verwendbar

es bleibt aber das allgemeine Internet-Problem, es gibt keine Übertragungs- und / oder Bandbreiten-Garantie (Best Effort (nur: beste Bemühungen (der Internet-Provider)))

viele Risiken (Beschädigung von See-Kabeln, Ausfall von Satelliten, gestörte Glasfaser-Leitungen unter meterhohem Schnee und Eis, ...)

Reparaturen / alternative Datenwege nicht planbar

### Kombination von ....:

- MPLS – Internet
- MPLS – LTE od.ä.
- Internet – Internet (mit unterschiedlicher Güte)
- Internet – Internet – LTE
- 

zur Optimierung der Stabilität arbeiten viele Nutzer mit 2 unabhängigen MPLS, fällt ein System aus, ist das andere ev. noch nutzbar

bedeutet aber auch doppelten Aufwand / Kosten

als Alternative wird nun eine MPLS durch das Internet ersetzt und SD-WAN genutzt durch beschleunigt sich die Anbindung neuer (abgelegener) Standorte nochmals, da nur das Internet als Basis (Underlay) gebraucht wird (hier ist dann zuersteinmal auch kein MLPS als alternativer Weg verfügbar)

als Backup lässt sich LTE od. ä. verwenden, für den Fall, dass Kabel beschädigt werden od. ä.

---

### Anwendung:

- Ladenketten
- Außenstellen
- Versicherungs-Büro's od. ä.
- mobile Büro's auf Baustellen
- Events
- Teleworker
- zukünftig vllleicht für Krankenwagen, Polizei-Einsatzfahrzeuge, ...
- ...

da nur Internet-Anschluss gebraucht wird, ist eine Einbindung innerhalb weniger Minuten möglich

### Anforderungen an eine SD-WAN-Lösung (lt. Gardner)

- **Transport Agnostic**                      Unabhängigkeit vom Netz-Medium / -Typ
- **Dynamic Path Selection**            Möglichkeit der Verteilung von App's auf verschiedene Leitungen  
(mit Policy Routing ist das bei LISP (→ [7.10. Locator ID Separation Protocol - LISP](#)) möglich)
- **Simple Interface**                      leichte Bedienoberfläche für Administratoren  
bei LISP (→ [7.10. Locator ID Separation Protocol - LISP](#))  
nicht vorhanden
- **VPN**                                        verschlüsseltes Virtuelles Netzwerk

derzeit aktuelle Technik mit Potential

gute Nachfrage von Kunden

viele große Anwender

Kosten-Frage

Personal-Frage (es gibt zu wenige ausgebildete Administratoren, ...)

verbesserte Automatisierung (Personal-Einsparung, Verringerung der Fehler, ...)

ständige Vergrößerung der benötigten Daten-Mengen

relativ gut erweiterbar

relativ schnelle Anpassungen von der technischen Seite notwendig

veränderte Unternehmens-Strukturen

Verknüpfung von Behörden, ...

### 7.9.1. Locator ID Separation Protocol - LISP

Möglichkeit zur Implementierung eines SD-WAN

offener Standard (derzeit selten umgesetzt)

hat nichts mit der Programmiersprache LISP zu tun

Trennung von Standort (Location) und Rufnummern / Adresse (ID)

ID bleibt unverändert, wählt sich aber bei unterschiedlichen Netzen etc. ein ((temporäre) IP wird immer neu zugeordnet)

---

Lösungen sind zwei IP-Nummern-Kreise

- für den Ort (Location) → den Routing Locator (RLOC), der sich ändern kann
- für die ID → den Endpoint Identifier (EID), der unveränderlich ist

für RLOC und EID sind jeweils unabhängig voneinander IPv4 und IPv6 möglich

RLOC

- ist die öffentliche IP unseres lokalen Netzwerkes
- somit das Routing-Ziel der LISP-Pakete
- ändert sich recht häufig, z.B. durch:
  - DSL-Zwangstrennung
  - Rooming (LTE, ..., WLAN)
  - mobile Dienste
  - Umschaltung auf Ersatz- oder Reserve-Leitungen
  - Übergang von Mobilfunk auf WLAN (Medienbrüche)
  - ...
- ...

EID

- IP im Nutz-Netzwerk
- allen Nutzern wird eine solche IP zugewiesen
- ev. nur die Firewall vor einer DMZ (Demilitarisierte Zone), in dieser wird mit NAT gearbeitet

das Mapping System bringt nun RLOC und EID zusammen

- Router registriert seine EID
- verwendet wird ein Map Server (Datenbank) → beinhaltet aktuelle Zuordnung von RLOC und EID
- der Map Resolver (MR, -Auflöser) fragt den MS ab
- Funktionsweise recht ähnlich zu DNS
- MS und MR laufen oft auf dem gleichen Router (Server)

Router-Typen für LISP:

- iTR → (ingress Tunnel Router) Router nur für eingehenden Verkehr
- eTR → (egress Tunnel Router) Router nur für ausgehenden Verkehr
- xTR → (Tunnel Router) Router nur für ein- und ausgehenden Verkehr gemeinsam

Umsetzung von:

- AVM
- Cisco
- Linux → Open Overlay Router

mit einem Internet Service Provider

oder

mit z.B. zwei ISP

benötigt aber auch zwei Router (empfohlenes Netzwerk-Design)  
diese Konstellation lässt auch eingehendes Load-Balancing zu (für Verbindungs-Zahlen)  
der Administrator legt dazu für jeden Nutzer ein Verhältnis für die Belastung der Leitungen fest (bezieht sich aber auf die Anzahl der Verbindungen, nicht auf die Datenmenge (da diese ja vorneweg unbekannt ist))

---

LISP lässt hinter einem xTR mehrere EID's zu, diese werden Instanzen genannt  
jede Instanz hat eine eigene und eineutige ID  
die Instanzen sind mit jeweils eigenen VRF verknüpft, so dass auch unabhängige Routing-  
Tabellen für die unterschiedlichen EID-Instanzen existieren

### **7.9.1.1. Proxy-xTR**

ist die Verbindung ins Internet  
Arbeits-Verfahren (ausgehende Daten):

- ? ist IP-Adresse bekannt
  - WENN ja, DANN: (Adresse liegt hinter einem RLOC) bauen des LISP-Paketes und senden an den anderen xTR
  - SONST: (da dann Ziel im Internet liegt,) bauen des LISP-Paketes und an Proxy-xTR senden

bei eingehenden Daten (Proxy fragt):

- kommen die Daten von einem registrierten RLOC
  - WENN ja, DANN: weiterleiten an betreffenden xTR
  - SONST: Paket verwerfen

Verschlüsselung gehört zu LISP

bei Cisco nennt sich das GetVPN

Schlüsselaustausch über einen Schlüssel Server (Key Server)

Verschlüsselung über IPsec

keine Point-to-Point-Verschlüsselung, sondern nur Verschlüsselung des inneren Paket

für jede Instanz ist eine extra Verschlüsselung möglich

für IPv4 und IPv6 ist Verschlüsselung ebenfalls unabhängig voneinander

### **7.1.1.2. Bewertung von LISP**

eingeschränkt für SD-WAN geeignet (lt. Gardner-Anforderungen)

bei Cisco ohne extra Lizenz dabei

die Linux-Implementierung muss noch geprüft werden

gut für statische Umgebungen (wenige neue / mobile Standorte) geeignet

interessante Alternative, offener Standard

## **7.9.2. Viptela**

von Cisco

---

Viptela war Start-Up, welches ein einfaches SD-WAN-System zur Vernetzung von Standorten zum Ziel hatte

2017 von Cisco für über 600 Mio. Dollar gekauft

andere Systeme anderer Firmen sind ähnlich

neben Meraki, LISP (→ ) und iWAN ist Viptela das vierte System bei Cisco, um SD-WAN umzusetzenderzeit LISP noch unvollständige Lösung, iWAN wird nicht weiter betreut

	<b>Viptela</b>	<b>Meraki</b>
<b>Unterschiede</b>	Enterprise Grade SD-WAN nur Vernetzung von Standorten komplexer, schwieriger zu bedienen mehr Möglichkeiten läuft auch auf Cisco-Routern	für kleine Unternehmen volles Sortiment für Vernetzung Firewall mit SD-WAN Switches WLAN-Access Point's Kamera
<b>Gemeinsamkeiten</b>	läuft jeweils auf eigener Hardware	

#### Viptela vEdge

- virtuelle oder physikalische Router an den Standorten
- verschiedene Modelle
  - Uplinks
  - Bandbreite
  - Leitungs-Arten
- ...

#### Viptela vManage

- Management Tool
- Single Point of Contact (zentrale Adminstartions-Stelle)
- Zugriff per API (automatisierbar)
- Cloud Application
- ...

#### Viptela vBond

- Erstellung der initialen Verbindungen
- Einbindung neuer geräte
- lässt sich redundant anlegen
- ...

#### Viptela vSmart

- übernimmt Konfiguration von vManage
- macht die eigentliche Konfiguration der Geräte
- ...

fällt die Controller-Struktur mit vManage, vBond und vSmart aus, dann läuft das System ersteinmal weiter

da keine neuen Geräte oder mal abegtrennte nicht mit einer Konfiguration versehen werden können, fällt das Gesamtsystem dann sukzessive aus.

Normalerweise ist Wiederherstellung der Strukturen deutlich schneller als die Ausfall-Geschwindigkeit

---

### **7.9.2.1. Zero Touch Provisioning**

neue Geräte werden ohne den direkten Eingriff oder Vorbereitung (an der Hardware) ins Netzwerk integriert

Gerät wird per Post od.ä. versandt

nach dem Einstöpseln ins lokale Netzwerk bekommt das Gerät eine lokale IP und das Gateway zugewiesen

dann wird der ZTP-Server (Zero Touch Provisioning-Server; Viptela od.ä.) kontaktiert  
es folgt die Registrierung des Gerätes und es wird in vManage angezeigt

Gerät erhält die vom Administrator vorbereitete Konfiguration übers Internet  
Nutzung von Templates möglich, da viele Optionen immer gleichartig sind  
nur z.B. die SD-WAN- bezogene IP-Nummer ändert sich

### **7.9.2.2. ausgewählte Feature's von SD-WAN und zugehörigen Lösungen**

#### **Zuordnung von Applikationen / Diensten zu bestimmten Leitungen / Leitungstypen**

unterschiedliche Anforderungen von den Applikationen. z.B.:

- Voice (wenig Bandbreite, hohe Güte (keine Verluste hinnehmbar) → MLPS
- File Transfer (hohe Bandbreite; Zeitverzögerungen praktisch kein Problem, Ausfälle werden durch das Protokoll selbst korrigiert) → Internet

unterschiedliche Laufzeiten von Signalen

- MLPS: maximale Bandbreite, Latenzen
- Internet: Best Effort, veränderliche Werte

Beachtung von Leistungs-Ausfällen

- Welche Applikation braucht ständige Verbindung? Kommt es zum Applikations-Stop beim Leitungs-Ausfall?
- beim Internet kommt es i.A. nur zu verzögerten Antwortzeiten
- Welche Applikationen können, in welcher Reihenfolge, abgeschaltet werden, um den obligatorischen Datenverkehr aufrecht zu halten?
- Welche Applikationen können mit reduzierter Bandbreite arbeiten?
- Kann mit zusätzlichen Maßnahmen (z.B. Komprimierung, Verzicht auf Verschlüsselung) die Leitung besser ausgenutzt werden

muss ständig überprüft und aktualisiert werden

#### **Anpassung von Standorten usw.**

- neue Filialen
- neue Rechenzentren
- neue Technik
- Aktualisierung der Software, ...
-

---

## **8. Netzwerke und Protokolle am Beispiel "Internet"**

basierend auf den open-hpi-Kurs "50 Jahre Internet – Internetworking 2019" Okt.-Dez. 2019 von Prof. MEINEL  
neu geordnet, selektiert und erweitert

heute rund die Hälfte der Weltbevölkerung ist online  
in Deutschland fast 100%

was passiert heute in 1 min im Internet → die **Internet-Minute**

- 3,8 Mio Suchanfragen gestellt
- 1 Mio Streams angesehen
- 2,1 Mio. Snaps erstellt
- 188 Mio. eMails versendet
- rund 400'000 Apps heruntergeladen
- 1 Mio. Login's
- 4,5 Mio Videos angesehen
- 1,4 Mio. Mal gewischt (Wechsel zwischen Anwendungen)
- rund 50'000 neue Beiträge erstellt
- rund 42 Mio. Nachrichten gesendet

(lt. statista.com)

### **8.0. Einleitung**

#### **8.0.x. Digitalisierung**

gemeint ist eigentlich die Übertragung von analogen / nicht-computerisierten Sachverhalten und Prozessen in die digitale / Computer-basierte Welt

z.B. Musik, Bilder, Video's

heute ist damit die Veränderung von Wirtschaft, Handel, Politik, Kultur, Bildung und Gesellschaft hin zu einer Digital-Technologie → digitale Transformation der Gesellschaft  
4. große Revolution in der Menschheits-Geschichte

Veränderung der Kommunikation der Menschen (Chat, Messenger, Social Media, ...)  
viel mehr verfügbare / auswählbare Information

praktisch immer mehr Dinge / Arbeiten der "analogen" Welt werden digital umgesetzt  
digitale Notizen  
Planungen / Termin-Absprachen  
Video-Konferenzen  
digitale Ticket's  
Homeoffice



---

## neue Möglichkeiten

- Navigations-App's
- Bezahl-Systeme
- Buchungs-System
- Telemedizin
- Smarthome
- autonomes Fahren
- Unfall-Assistenten
- Webinare
- Fernsteuerung von Industrie-Anlagen
- neue Arbeitsplätze / Arbeits-Techniken (agil + kooperativ)
- digitale Wahlen / Meinungs-Umfragen / online-Petitionen
- Verfolgung von Postsendungen
- digitale Verwaltung
- 

## Vorteile / PRO-Argumente

- hohe Aktualität
- mehr soziale / politische Möglichkeiten
- weltweite / breitere Verfügbarkeit von Daten usw.
- viel mehr Möglichkeiten der Nutzung von Dingen, Prozessen und Daten
- mehr individuelle Freiheit bei geringerer staatlicher Aufsicht
- mehr Kreativität durch Gestaltung von Medien
- Gaming, online-Gaming
- neue Arbeits-Methoden
- leichtere Reaktion auf Entwicklungen / Veränderungen / ...
- ständige Erreichbarkeit
- internationaler Handel / Verkauf von Waren
- 

## Nachteile / KONTRA-Argumente

- Verringerung der direkten (Face-to-Face-)Kommunikation; Ablösung durch indirekte (gefühl-ärmere / weniger empatische) Kommunikation
- die große Menge an auswählbaren Informationen führt zur Orientierung auf das eigene Nachrichten-Universum
- Fake News / Desinformation / "Alternative Fakten"
- Hate speech
- eingeschränkte Umwelt-Orientierung (Aufnahme / Erkennen von Umwelt-Situationen (Verkehr, ...))
- Daten können (einfacher) mißbraucht werden / Tracking
- Schutz der Personendaten / gläserner Mensch / Verlust/Einschränkung der Privatsphäre
- Urheberrechte schwerer durchzusetzen
- permanente Erreichbarkeit / Verpassen-Angst
- Grenzen zwischen Arbeit und Freizeit / Urlaub verschwinden
- Cyberkriminalität; Spam, Computer-Viren usw.
- Nichtteilnahme erzeugt sozialen Druck oder Isolation (abgehängt sein)
- Cybermobbing; Burn-out
- gefühlter rechtsfreier Raum
-

---

neue Regeln sind notwendig  
Anpassung von gesetzlichen Rahmenbedingungen an die Geschwindigkeit der digitalen Entwicklung  
Digitalisierung lässt sich nicht aufhalten aber gestalten

## **8.1. kleine Geschichte des Internet's**

### **8.1.0. Kommunikation vor dem Internet**

ständig steigende Menge an Informations-Inhalten

mehrere Entwicklungs-Stufen

**zuerst dauerhaftes Festhalten als Wandmalerei und Schrift**

Symbol-Schriften (Wandmalerei vor rund 40'000 Jahren)

ungefähr 3'500 v.u.Z.

phonetische Symbol-Schrift → Keil-Schrift in Mesopotanien

3'000 v.u.Z.

Hieroglyphen (Bilder-Schrift) in Ägypten

enthielt Wort-, Silben- und Einzen-Konsonanten-Symbole

600 v.u.Z.

erste Grammatiken

Nachrichten-Agenturen ab 1848 (Associated Press) und 1851 (Reuters)

### **Orts-unabhängiges Festhalten**

unbewegliche Steine als Träger-Medium

dann (gebrannter) Ton als schon bewegliches Stein-Material, was aber meist verbaut wurde

auch Holz-Platten

dann ab v.u.Z. Papyrus und v.u.Z. Pergament

als Vorläufer des heutigen Papier's

erste große Sammlungen (Große Bibliothek von )

Erfindung des Papier's um 105 u.Z. in China

in Europa im frühen Mittelalter (12. Jhd.) nachentdeckt

### **immer größere Distanzen können überwunden werden**

ansonsten akustische Nachrichten-Übertragung per Ruf-Zeichen (Posten-Ketten)

---

oder auch Trommel-Telegraphie (z.B. Afrika, Nordamerika, Australien)  
dort z.B. auch das Did...du für größere Entfernungen

parallel die optische Signal-Übertragung  
per Handzeichen, Feuer- oder Rausch-Zeichen  
befördert durch die Entwicklung des Fernrohres (1609)

mittels Semaphore Signal-Ketten (17. Jhd.)  
erste größere Verbindung zwischen Paris und Lille (270 km, Relais-Stationen)  
daraus entstand dann landesweites Telegraphie-System, bestand bis 1853

abgelöst durch elektrisches Telegraphie-System  
ab 1730 durch GRAY  
1804 Elektrolyt-Telegraph mit 26 Glas-Röhrchen (Signal-Anzeige durch gebildete Gas-Bläschen)  
1820 elektromagnetischer Nadel-Telegraph von AMPERE  
Zeiger-Telegraph von GAUß und WEBER ab 1833  
Benutzung einer binären Signal-Übertragung

1837 MORSE-Alphabet  
direkt-ablesbarer Fernschreiber (Ticker) von HUGHES

beschrieben auch schon Brieftauben vor über 4'000 Jahren im alten Ägypten  
hier war die Übertragungs-Geschwindigkeit mit 60 km/h schon beachtlich groß

Boten- und Stafetten-Dienste  
ab 500 v.u.Z.

Lauf von Marathon nach Sparta (490 v.u.Z.; für 42 km brauchte der Läufer damals 2 Tage)

ab 1490 moderner Postdienst unter König Maximilian I.  
zwischen Mecheln (bei Brüssel) und Innsbruck  
betrieben vom Adelsgeschlecht Thurn und Taxis

deutsches Postwesen seit 1597  
um 1700 rund 20'000 Kuriere unterwegs

erster Schiffs-Postdienst in Europa ab 1633 zwischen Dover und Calais

erste Kabel-Verbindung (für elektrische Telegraphie) zwischen England und dem Kontinent um  
1851  
Kabel zwischen Irland und Neufundland (1956)

Glasfaser-Kabel (zur optischen Übertragung per Laser-Signal)

aktuelle Stufe ist eben das Internet

---

## 8.1.1. Computer als Voraussetzung für moderne Kommunikation

### *Geschichte der Computer-Technik*

ZUSE 1937 erster Programm-gesteuerter Rechen-Automat Z1, noch mechanisch

1939 – 40 erster Großrechner "Harvard Mark I" (kurz "Mark I") von AIKEN

Z3 von ZUSE dann 1941 auf Relais-Basis (elektromechanisch)

1942 – 43 entstand die Rechen-Anlage (elektromechanisch) "Colossus" in Betchley Park zur Entschlüsselung der Enigma

erster vollelektronischer Universal-Rechner "ENIAC"1945 mit 18'000 Röhren (1 Röhre entspricht ungefähr 1 Transistor) entwickelt von ECKERT, MAUCHLY, GOLDSTINE und VON NEUMANN

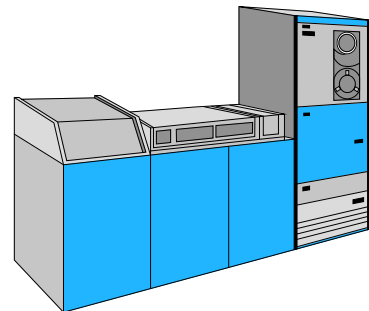
1947 Erfindung des Transistors

durch Miniturisierung "Integrierte Schaltkreise" (IC ... integrated ) 1958 mit mehreren Transistoren zusammen als eine Funktions-Einheit

→ DATAPROS.CGM

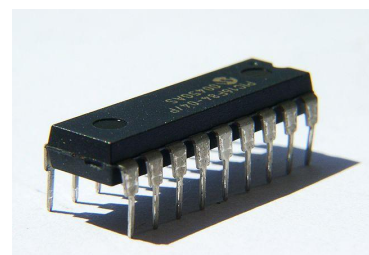
1956 Magnet-Platten-Speicher durch IBM

Anfang der 60er Jahre dann erste "Minicomputer" in Schreibtisch- bis Koffer-Größe auf der Basis von Integrierten Schaltkreisen Leistung-Fähigkeit entspricht bis hier praktisch nur heutiger Taschenrechner vorrangig Wert auf Miniturisierung gelegt wenige wirklich Großrechen-Aufgaben vorhanden



treibende Kräfte für Leistungs-Steigerung:

- Militär (Atombombe, Simulationen von Atombomben-Explosionen, Versorgungs-Planung, ...)
- Geheimhaltung (Kryptographie (Ver- und Entschlüsselung))
- Metreologie (Wetter- und Klima-Modelle)
- 



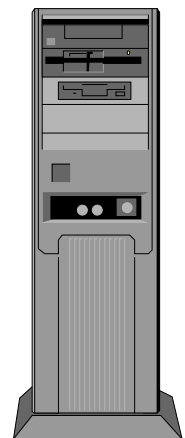
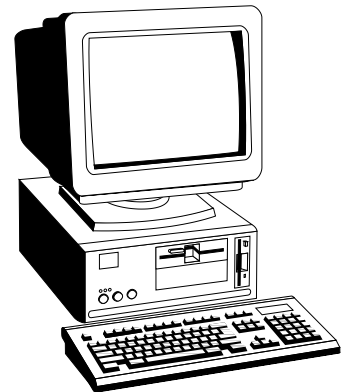
integrierter Baustein  
Q: de.wikipedia.org (Wollschaf

---

1970 erster Mikroprozessor Intel 4004 mit 4 bit Verarbeitungs-Breite (heute 64 oder 128 bit üblich) mit 2'300 integrierten Transistoren auf praktisch einem Chip



erster PC 1975 von Apple (JOBS, WOZNIAK) genannt "Apple II"  
erst 1981 erster "IBM-PC" mit vielen standardisierten Bausteinen / Zusatzkarten für die individuelle Gestaltung genannt "XT"



→ CRAY.CGM

---

## Computer-Generationen

- **1. Generation (1945 – 1956)** spezielle Anwendungszwecke / Nutzungs-Szenarien  
Programmierung / steuerung direkt über Maschinen-Befehle (sehr individuell für jede einzelne Maschine)  
Röhren- bis hin zu Transistor-Technik  
Datenträger sind Lochkarten, Lochbänder  
zum Ende hin Magnetplatten-Speicher (Winchester-Platten)  
Größe Zimmer bis Schrank (selten Schreibtisch)
  
- **2. Generation (1956 – 1963)** erste freiere Programmierung (Assembler (Worte / Kürzel statt Dual- od. Oktal-Zahlen (Programme nutzen Stapel-Betrieb (z.B. Lochkarten-Sätze (praktisch nur Sequenzen!))  
Transistoren haben sich durchgesetzt  
Integrierte Bausteine zuerst als kleine Leiterplatte-Stücken (mit recht vielen zusammengehörenden elektronischen Bauelementen), dann erste echte Integrierte Schaltkreise (z.B. Gatter-Netze, Halb- und Voll-Adder)  
dazu nun Magnet-Bänder, Ferrit-Kern-Speicher  
Größe Schrank bis Schreibtisch
  
- **3. Generation (1963 – 1971)** verbreiter Einsatz durch integrierte Schaltkreise  
erste (auf mehreren Maschinen nutzbare) Betriebssystem mit Mehr-Programm-Betrieb  
erste höhere Programmiersprachen (für Profi's) ALGOL,  
Größe Schreibtisch bis Koffer
  
- **4. Generation (1971 – ...)** individuelle Nutzung / Programmierung mit höheren Programmiersprachen (z.B. BASIC, )  
hoch-integrierte Schaltkreise mit Mrd. von Transistor-Äquivalenten Mikroprozessoren als Zentraleinheit → Mehrprozessor-Systeme  
Größe Schuh-Karton  
interne und externe Vernetzung (Intra- und Inter-Net)  
starker Preisverfall → Home-PC  
Parallelisierung von Programm-Abläufen
  
- **5. Generation** Licht- und / oder Quanten-Computer  
Hologramm-Speicher, Kristall-Speicher  
weitere Entwicklungs-Tendenzen:
  - künstliche Intelligenz (KI)
  - maschinelles Lernen
  - Ubiquitous Computing (Überall-Computer)
  - Sprach-Erkennung
  - weitere Miniturisierung (→ Smartphone)
  - Cloud-Computing
  -

Groß-Rechenanlagen blieben auf der Größe von Räumen / Hallen stehen, allerdings stieg die Leistung sehr stark

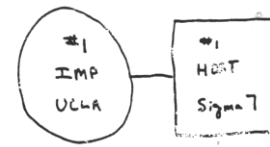
## 8.1.2. Entstehung des Internet's

29. Oktober 1969 war der Start dessen, was wir heute Internet nennen

4 Computer an 4 Universitäten (Los Angeles, Santa Barbara, Stanford, Utah) → ARPANET

Ziel war Verbindung von Rechnern mit verschiedenen Systemen

Geräte-Verbund über sogenannte IMP



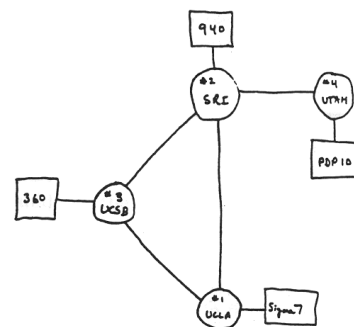
THE ARPA NETWORK

SEPT. 1969

1 NODE

FIGURE 6.1 Drawing of September 1969 (Courtesy of Alex McKenzie)

erster Service war telnet (Einloggen auf einem entfernten Rechner; Fernsteuerung / Fernbedienung von Rechnern auf Kommandozeilen-Ebene)



THE ARPA NETWORK

DEC 1969

4 NODES

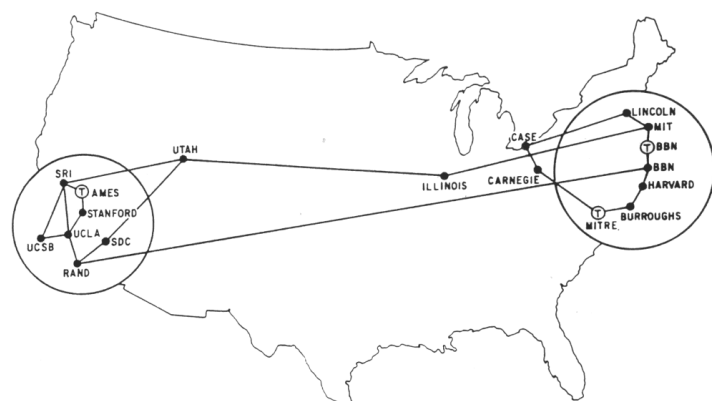
FIGURE 6.2 Drawing of 4 Node Network (Courtesy of Alex McKenzie)

1971 schon 23 Host's mit 15

Knoten im ARPANET

erste eMail versandt (erst 1983 erste in Deutschland)

ALOHAnet als erster Funk-Netzwerk zwischen hawaiianischen Haupt-Inseln



MAP 4 September 1971

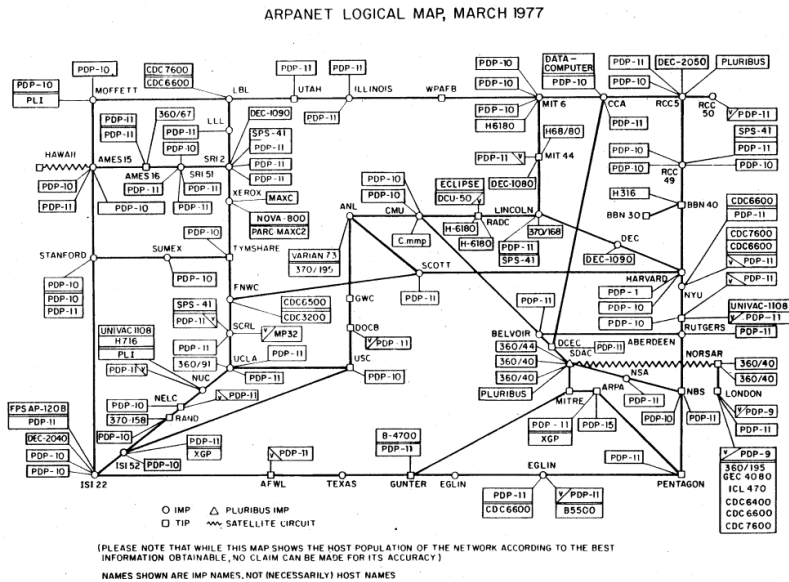
1972/73 KAHN und CERF entwickeln weiträumigen Rechner-Verbund bei der DARPA

1973 Einbeziehung von England und Norwegen in das Netz  
Internet bestand aus ungefähr 500 Host's

Ethernet-Technologie (METCALFE und BOGGS)  
neues Modell zur Netzwerk-Kommunikation mit mehreren Abstraktions-Schichten zur Aufgaben-Teilung; beschäftigt sich vorrangig mit der Adressierung der Geräte, dem Daten-Transport und der Verbindungs-Vermittlung (heute TCP/IP-Schichten-Modell)

1974 Geburtsstunde des TCP (Internet Transport Control Program) (hier noch ein anderes TCP), mehr eine Implementierung von Algorithmen für die Internet-Arbeit; nicht direkt vergleichbar mit dem heutigen TCP (Transport Control Protocol)

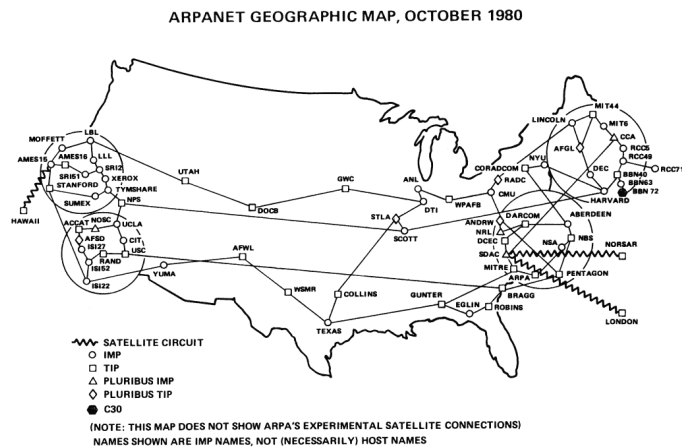
1976 IP-Router (STRAZISAR)



1978 Vollzug der Auftrennung in die Schichten TCP (Transport-Schicht) und IP (Internet-Schicht)

??? FTP  
??? weitere Internet-Service's / Internet-Anwendungen

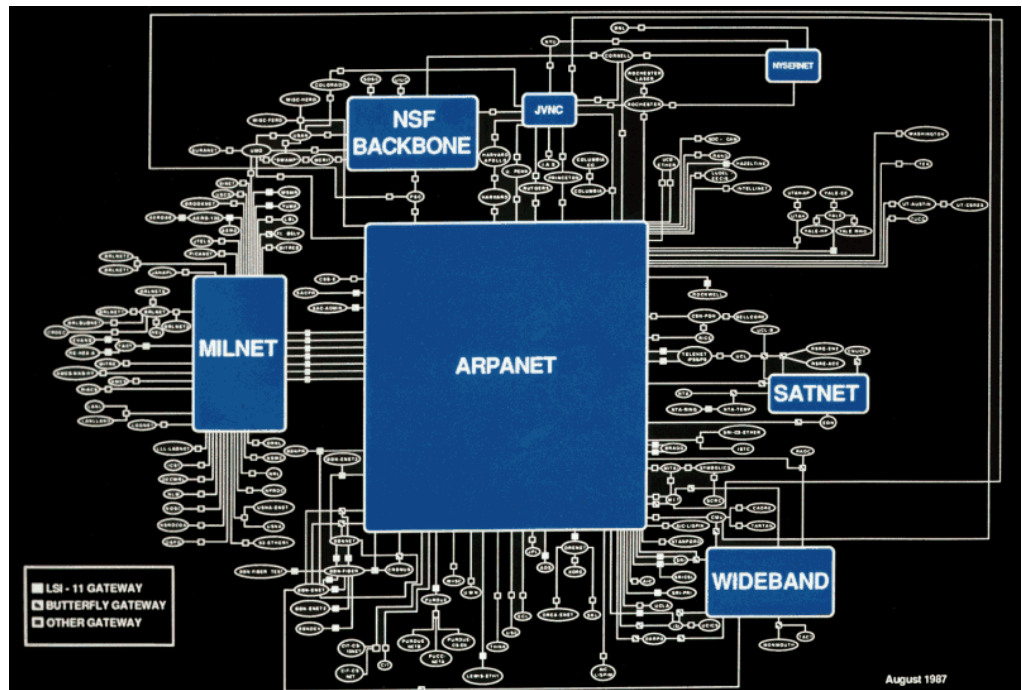
1980 Ethernet mit 10 Mbit/s verfügbar (meist in Bus-Topologien)  
Einführung der WLAN's nach IEEE 802.11 (entgeltliche Verabschiedung 1997; deshalb damals auch schwierige Kopplung der Geräte und Netze)



1981 RFC 791 beschreibt das heute weit verbreitete IPv4

1983 im ARPANET vollständige Umstellung der Adressierung auf TCP/IP (→ IPv4)  
damit praktisch die echte Geburtsstunde des heutigen Internet's  
Urväter CERF und KAHN  
Aufspaltung in ziviles, öffentliches Netz (weiterhin ARPANET) und ein relativ unabhängiges militärisches, geheimes Netz (MILNET)





Netze um 1987

nächster Entwicklungs-Schub war die Verfügbarkeit von PC's (1975 bzw. 1981)

1985 erste Domain registriert (nordu.net) Einführung des DNS-Dienstes zur Übersetzung von Domain-Namen in nutzbare IP-Adressen  
 rund 2'000 Host's im Netz  
 Netzwerk-Bridge als Geräte-Klasse eingeführt



ab 1986 wird die Anbindung aller Universitäten der USA gefördert

1988 erster Internet-Wurm (betroffen sind 10 % der 60'000 angebotenen Host's)

---

1989 rund 150'000 im Netz  
Einführung des www / http durch BERNERS-LEE vom europäischen CERN  
entwickelte HTML als Seiten-Beschreibungssprache  
1990 erster Web-Browser zum Anzeigen von HTML-Seiten und einer passenden Server-Software (für Unix-/Linux-Rechner) durch VAILLIAU und BERNERS-LEE ("Geburtsstunde des www)

wichtigster Service ist www / http  
hat die Verbreitung des Internet's für den "Normal"-Nutzer in Gang gesetzt

1990 Still-Legung des veralteten ARPANET's  
erste (noch heute besuchbare web-Seite <http://info.cern.ch/hypertext/WWW/TheProject.html>)  
große EtherSwitches (von Kalpana) zur Verbindung von großen Ethernet-Domains

1993 erster Browser mit graphischer Benutzer-Oberfläche "NCSA Mosaic"  
damit Internet (www) für die breite Masse nutzbar  
heute rund 1 Mrd. Rechner im Internet (über DNS ansprechbar)  
weltweit rund 4,5 Mrd. Internet-Nutzer  
mit 20 Mrd. Web-Seiten (extrem dynamisch)

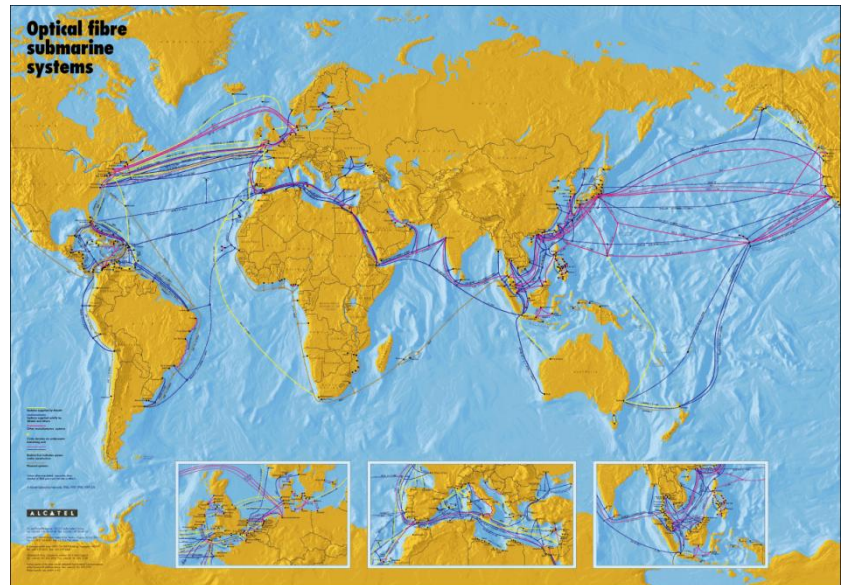
1994 Standardisierung der Dienste über das W3C (World Wide Web Consortium)  
Gründung von Netscape (Vorläufer des FireFox-Browser's "Netscape Navigator" (war noch Kauf-Programm; aus Mosaic-Browser abgeleitet)) und amazon  
Zeit des Browser-Krieg's; jeder Hersteller versuchte besondere Feature's einzubauen, um sich gegen die Konkurrenz durchzusetzen  
Netscape führt SSL in der 1. Version ein; zusätzliche (optionale) Schicht zwischen der Transport- und der Anwendungs-Schicht

1995 wird mit Windows 95 der erste mitgelieferte Browser ("Internet Explorer") ausgeliefert  
Gründung vieler innovativer Firmen mit Internet-Thematiken (dot-com's)  
großer Hype an der Börse, Handel mit Ideen für's Internet, aber noch wenig Umsetzung, dot-com-Blase platzte dann 2001  
Ethernet mit 100 Mbit/s nun vorrangig als Stern-Topologie (vorrangig an Hub's, später dann Switche)

große Suchmaschinen waren zu dieser Zeit Fireball, Altavista, yahoo

1998 Gründung von Google mit neuem Ranging-Algorithmus () und besonders einfacher Start-Seite  
heute dominierende Suchmaschine  
mehrere weitere Unterbereiche (Firmen) im Konzern

1999 Ethernet mit 1 Gbit/s  
vorrangig als Backbone  
gedacht (heute im norma-  
len LAN Standard)  
WLAN nach IEEE 802.11a  
mit 11 Mbit/s  
Standardisierung der In-  
ternet-Sicherheit aus SSL  
V.3.1 wird TLS 1.0



interkontinentale optische Netzwerk-Verbindungen

2001 Gründung des wikipedia-Projekt's  
Ethernet mit 10 Gbit/s praktisch nur für professionelle Zwecke , z.B. als Backbone (Hard-  
ware und Verkabelung sehr anspruchsvoll und teuer)

2003 Gründung von facebook (ZUCKERBERG)

2005 Wandlung vom Web 1.0 (dem Angebot's-Netz bisher zum Mitmach-Netz Web 2.0)  
youtube (Musik- und Video-Streaming)  
social web, semantic web, service web

2007 erste verbreite Smartphone's (iPhone von apple)

2014 Orientierung auf mobile web  
zusätzliche Ethernet-Version mit 2,5 und 5 Gbit/s mit vereinfachter HardwVre und verkabe-  
lung

ab 2015 verbreitet sich IoT (Internet of Things) sehr stark

??? VoIP

??? Cloud's / Cloud-Computing

2018 Einführung / Standardisierung von TLS 1.3

2019 neueste WLAN-Standard's für 11 Gbit/s nach IEEE 802.11ax&ay

---

### **Web-Entwicklungs-Stufen**

- **Web 1.0**  
**Web of Content**                      statische Internet-Seiten  
einfache Graphik, wenig Multimedia  
geringe Seiten-Größen und Auflösungen
- **Web 2.0**  
**Web of Communication**            Multimedia  
interaktives Web  
dynamische Web-Seiten  
soziale Netzwerke
- **Web 3.0**  
**Web of Context**                      semantische Web  
personalisierte Web-Seiten  
Verkaufs-Vorschläge ("andere Nutzer haben auch ge-  
kauft ...")
- **Web 4.0**  
**Web of Things**                      Einbindung von Geräten, Sensoren, Aktoren ins Netz  
Smarthome (Haus-Automation),  
Optimierung auf mobile Geräte  
virtuelle Netzwerke  
Geo-Tracking  
virtuelle Realität
- **Web 5.0**  
**Web of Thoughts**                    Anwendungen der Künstlichen Intelligenz  
Steuerung über Blicke, Gedanken, ...

### 8.1.3. Smartphone's als neue Dimension der Internet-Nutzung

angefangen hat es mit dem Nokia 9000 Communicator (15. August 1996)  
damals war Europa – hier speziell Norwegen – noch das Zentrum der Handy-Innovationen  
als "Büro in der Westentasche" (Preis 2'700 DM)  
Telefon + FAX + Adressbuch + Taschenrechner + Notizblock + Internet-Browser

Multimedia und Apps kaum möglich, es fehlte die Bandbreite

WAP Wireless Application Protocoll

ab 1997

sorgte für bessere Ausnutzung der kleinen Display's (meist noch einfache Hanfy-Display's,  
vielfach sogar noch einfarbig)

geringe Datenmengen und kleinere Ladezeiten → geringere Kosten (noch getaktete Abrech-  
nungs-Modelle)

erstes iPhone 09. Januar 2007

erstes echtes Smartphone im heutigen Design; Touchscreen !!!

intuitive Bedienung

Telefon + Mediaplayer + Kalender + eMail + Notizen + ... + weitere Apps möglich → mobiles  
Kommunikations-Gerät

---

heute immer mehr Sensoren, größere Display's, Digital-Kamera, ...  
App's bringen heute die wesentliche Funktionalität der Smartphone's aus

nach dem Ort der Datenverarbeitung unterscheidet man heute die

### **App-Typen**

- **native App's** für das konkrete Betriebssystem entwickelt  
haben Zugriff auf Hardware-Ressourcen und Betriebssystem-Funktionen  
Installation übers Internet (aus App-Store)  
für den Betrieb an sich keine Internet-Verbindung notwendig  
oft Internet für die Nutzdaten gebraucht  
  
Wartung / Aktualisierung / ... u.U. aufwendig  
u.U. eingeschränkte Nutzung ohne Internet möglich
- **web-App's** Nutzung über Internet (im Browser)  
unabhängig vom Betriebssystem, aber von einem aktuellen Browser  
keine Installation notwendig  
Wartung / Aktualisierung / ... erfolgt auf dem Server des Anbieters  
  
keine Nutzung möglich, wenn kein Internet verfügbar ist  
begrenzte Zugriffe auf Hardware-Ressourcen und Betriebssystem-Funktionen
- **hybride App's** Kombination der Vorteile aus den nativen und web-App's  
meist für mehrere Betriebssystem ausgelegt  
viele Funktionen brauchen kein Internet  
erweiterter Daten-Austausch übers Internet  
Daten-Verbindungen unabhängig vom Betriebssystem  
  
für die volle Nutzung muss Internet verfügbar sein

mit ihrem mobilem Zugang zum Internet war die Erfolgs-Geschichte erst möglich

heute praktisch hat jeder 2. Weltbewohner ein Smartphone od.ä.

negative Konsequenz → ständige Verfügbarkeit; steigender Streß

## 8.2. Rechnernetze als Basis des Internet's

### Bit's und Byte's

Daten-Kodierung erfolgt auf der Basis des Dual-System (also nur durch 0 und 1 (bzw. AN und AUS oder Strom / Spannung vorhanden und nicht-vorhanden))

bit (binary digit) ist eine Binär-Zahl od. auch Dualzahl

Kodierung muss standardisiert sein, damit Daten für alle Netzwerk-Beteiligten gleichartig verstanden werden

Kodieren und Dekodieren sind entgegengesetzte Teilfunktionen

Daten werden als Bit-Folgen kodiert

für Versand ist die Kenntnis der Daten-Kodierung nicht notwendig

hier meist spezielle Kodierung der Informationen 0 und 1 als Spannung oder Licht-Impuls usw. usf. notwendig; aber gleiches Prinzip, wie bei Daten-Kodierung

Bit's werden in physikalische Signale umgesetzt und so übertragen

Leitungs-Kodierung legt fest, wie ein Signal in der physikalischen Ebene (Schicht) übertragen wird

Ziel ist eine optimale und fehlerfreie Daten-Übertragung auf einem bestimmten (Übertragungs- / Kommunikations-)Medium

Ethernet

Kabel-gebunden, Kabel wird von allen Netzwerk-Teilnehmern genutzt

Ziel ist senden und gleichzeitigen Lesen (Lauschen) auf dem Medium (CSMA/CD als Verfahren)

Anfang und Ende einer Übertragung muss sauber identifizierbar sein  
einheitlicher Takt / Synchronisationen

Grund-Bit-Pakete sind 8 bit, die als Byte zusammengefasst

nach SI-Einheiten-System (für den Normalgebrauch)			nach IEC-Festlegung (für den informatischen Fachbereich)			Fehler [%]
dezimal Skalierung			binäre Skalierung			
Skalierungs-Faktor: $1'000 = 10^3$			Skalierungs-Faktor: $1'024 = 2^{10}$			
Wert	Abk.	Präfix	Wert	Abk.	Präfix	
$10^3 = 1'000$	k	Kilo	$2^{10} = 1'024$	Ki	Kibi	4,63
$10^3 = 1'000^2 = 1'000'000$	M	Mega	$2^{20} = 1'024^2 = 1'048'576$	Mi	Mebi	
$10^6 = 1'000^3$	G	Giga	$2^{30} = 1'024^3$	Gi	Gibi	
$10^9 = 1'000^4$	T	Tera	$2^{40} = 1'024^4$	Ti	Tebi	
$10^{12} = 1'000^5$	P	Peta	$2^{50} = 1'024^5$	Pi	Pebi	
$10^{15} = 1'000^6$	E	Exa	$2^{60} = 1'024^6$	Ei	Exi	

---

## binäre, dezimale und hexadezimale Zahlen-Darstellung

	zulässige Ziffern-Symbole	Zahlen-Basis	Beispiel (exakt)	häufige / alternative Notierung in der Informatik	
<b>binär</b>	0, 1	2	1110 <sub>2</sub>	1110b    0x1110	
<b>dezimal</b>	0, 1, ..., 9	10	14 <sub>10</sub>	14	
<b>hexadezimal</b>	0, 1, ..., 9, A, B, ..., F	16	E <sub>16</sub>	0xE	

für hexadezimale Symbole dürfen auch Kleinbuchstaben verwendet werden

## Netzwerke und Netzwerk-Typen

Netzwerk als Zusammenschluß von Geräten zum Daten-Austausch  
Geräte-Kommunikation

Netzwerk charakterisiert:

- Infrastruktur
- Daten-Pakte
- Adress-Formate
- Zugriffs-Steuerung
- ...

innerhalb eines Netzwerkes müssen die Netzwerk-Merkmale (Charakteristika) einheitlich sein

Voraussetzungen für ein Netzwerk:

- Kommunikations-Protokolle
- 

### **Netzwerk-Typen**

- **LAN**                                    lokale Netzwerke (z.B. Ethernet (bedeutenste Technologie))
- **WLAN**                                    kabellose Netzwerke (meist Funk)  
                                                  auch Ethernet
- **WAN**                                    Weitverkehrs-Netzwerke
- ...

### **Netzwerk-Komponenten**

- **Endgeräte** (Host's: Laptop's, PC's, Tablet's, Smartphone's, Server, Sensoren, Aktoren, ...)
- **Zwischen-Systeme / Infrastruktur-Komponenten** (Router, Hub's, Switches, Repeater, ...)
- **physikalische Verbindungen / Trägermedien** (Kabel, WLAN-Funk, Infrarot, ...)



---

Client-Server-Paradigma (→ Rollen im Netz)

- Server: Anbieter von Daten oder Leistungen; wartet auf Anfragen
- Client: Nutzer / Anforderer / Abfrager von Daten und Leistungen; startet Kommunikation mit Server

ein Server kann mehrere / viele Client's bedienen

ein Client kann an mehrere Server Anfragen (Request's) starten

Server überprüft Berechtigung (→ Autorisation) des Client's (Vorgang: Autorisierung)

### **Verbindungs-Typen**

- **Leitungs-Vermittlung** Verbindung über eine konkrete Leitung (Punkt-zu-Punkt-Verbindung)  
Länge der Daten ist frei von beiden Host's bestimmbar  
Verbindung besteht auch, wenn keiner der Host's sendet (→ beide Host's: lauschen)

vergleichbar mit:

- Telefonat

- **Paket-Vermittlung** Verbindung kann von beiden Host's genutzt werden  
auch andere Host's können diese Verbindung nutzen  
Daten werden in Pakete verpackt (haben bestimmte Länge und Struktur)  
auch andere (auch nicht berechnigte) Host's können mitlesen

vergleichbar mit:

- Paket-Dienst der Post

physikalische Medien unterliegen immer auch Störungen und Abschwächungen  
außerdem wird aus Effektivitäts-Gründen auch immer in der Nähe von Limit's gearbeitet  
deshalb auch immer Fehler-Erkennung und -Korrektur notwendig

für größere Distanzen sind Zwischen-Verstärker od.ä notwendig  
für komplexere Netze sind Verteiler der Daten notwendig

jeder Host verfügt über (mindestens) einen Netzwerk-Adapter  
dieser wandelt die Bit-Folgen zuerst in eine für das Netzwerk geeignete Form / ein Format  
und dann werden diese in physikalische Signale (für das Übertragungs-Medium) umgesetzt  
dann Übertragen der physikalischen Signale auf das Medium

### **Übersicht zu Netzwerken des Host's**

- **Windows** `ipconfig`
- **macOS** `ifconfig`
- **Linux** `ifconfig`



---

### ***Einteilung der Netze nach ihrer räumlichen Ausdehnung***

- **PAN**  
**Personal Area Network** bis um 10 m (Raum)
- **LAN**  
**Local Area Network** bis um 1 km (Gebäude, Firma, Campus)
- **MAN**  
**Metropolitan Area Network** bis um 10 km (Stadt, Ortsteil)
- **WAN**  
**Wide Area Network** 100 – 1'000 km (Land, Bezirke, Kreise)
- **GAN**  
**Global Area Network**  
**Internet** 10'000 km (gesamte Erde)

### ***Einteilung nach organisatorischer Abdeckung***

- **BAN / WBAN**  
**(Wireless) Body Area Network** Objekt- bzw. Person-bezogenes Netzwerk  
Smartwatch + Smartphone + Sensoren etc. in  
der Kleidung
- **PAN / WPAN**  
**(Wireless) Personal Area Network** persönliches / individuelles Netzwerk eines  
Raumes / Labor's / ...
- **LAN / WLAN**  
**(Wireless) Local Area Network** lokales Netzwerk eines Gebäudes

### ***Einteilung der Netze nach ihrer Aufgabe (Einsatz-Charakteristik)***

- **Funktions-Verbund** Netze aus Rechner mit spezieller Ausstattung / Periphe-  
rie / Datenstrukturen / Rechenleistung (Rechner-  
Cluster)... zusammengestellt wurden
- **Last-Verbund** Netze aus Rechner, die Belastungen (und Ausfälle) un-  
tereinander ausgleichen / verteilen
- **Daten-Verbund** Netze aus Rechnern, die gemeinsam die Daten beinhal-  
ten
- **Sicherheits-Verbund** Netze aus Rechnern, die bei Ausfällen die Aufgaben  
übernehmen (eigentlich untergeordnete Rechner über-  
nehmen nun die Aufgabe)  
z.B. sekundäre Domain-Controller, die einen primären  
Domain-Controller ersetzen können)

---

VPN (Virtual Private Network)  
(i.A. verschlüsselte) Punkt-zu-Punkt-Verbindungen über Paket-vermittelte Netzwerke

SAN (Storage Area Network)

### ***Einteilung der Netze nach ihrer Beschaffenheit / Strukturierung***

- **homogen** besteht aus gleichartigen Host's  
gleich Netzwerks-Muster
- **inhomogen** besteht aus verschieden-artigen Host's  
offene Netzwerke / Netzwerks-Muster
  
- **öffentlich** frei verfügbar; prinzipiell für alle Host's nutzbar
- **privat** nur für bestimmte Auswahl an Host's zugänglich

### ***Einteilung der Netze nach Verbindungs-Typen***

- 
- 
- 

### ***Einteilung der Netze nach ihrer Topologie***

- **Bus** alle Host's befinden sich entlang eines Verbindungs-Medium  
es kann immer nur ein Host senden (simplex), aber alle (anderen) empfangen die Daten gleichzeitig  
Kollisionen müssen ausgeschlossen werden (sonst Vermischung von Signalen)  
heute teilweise weniger problematisch durch Voll-Duplex-Leitungen
- **Ring** Verbindungs-Medium ist Ring-förmig ausgelegt  
Daten werden von einem Host zu einem Nachfolge-Host weitergereicht  
beim Ziel wird Paket aus Medium entfernt und Quittierung gesendet (über Bachfolge-Host bis zum ursprünglichen Sender)
- **Stern** Host's hängen an einem Zentral-Host (ev. kaskadiert)
- **Netz**
- **gemischt**

## Wer ist der Chef im Netzwerk?

Normalerweise stehen auf einem PC Daten und Anwendungs-Software im Speicher oder auf einem Datenträger bereit. Die Kommunikation läuft über den Bus. Der Prozessor kontrolliert und steuert den Bus.

So einfach geht es aber in Netzwerken nicht mehr. Hier sind die Rollen nicht so unkompliziert geklärt und auch nicht so star. Ein Gerät kann man Daten-Lieferant sein, mal Daten-Verarbeiter und dann mal wieder Daten-Empfänger. Mit den Eintritt weiterer Geräte ins Netzwerk können sich die Rollen dann schnell ändern. Man spricht auch von verteilten Anwendungen oder Systemen.

In den meisten Netzwerken und Protokollen nutzt man eine klare Rollen-Teilung.

Ein Gerät möchte Daten haben. Die Daten-Nutzer oder –Anfrager werden Client genannt. Ein Client stellt also eine Anfrage (Request). Diese Anfrage geht an der Server. Er stellt die Daten bereit und antwortet auf die Anfrage des Client's mit einem Reply. Diese Kommunikation kann sich beliebig oft wiederholen.

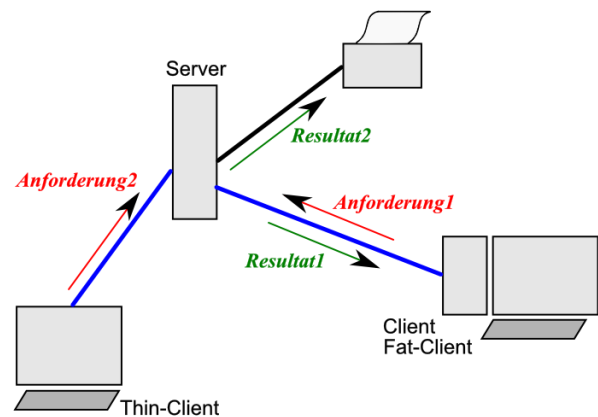
Allerdings verhält sich der Server mehr passiv. Er wartet auch einen Request (eine Anfrage). Nur dann wird er kurzzeitig etwas tun und mit einem Reply antworten. Die aktive Rolle kommt mehr dem Client zu. Der Client stellt die Anfrage und wartet dann auf die Server-Antwort, um diese dann weiter zu bearbeiten.

I.A. gibt es in einem Netz nur vereinzelte Server, aber viele Client's.

In der sparsamsten Version kann der Client nur die Ein- und Ausgabe-Kommunikation mit dem User. Wir sprechen dann von einem Thin-Client. Oft bestehen Thin-Client's nur noch aus Tastatur, Maus und einem Monitor sowie einer Netzwerk-Schnittstelle. Alle Berechnungen werden auf dem Server getätigt. Der Server ist praktisch der Chef des System's.

Fat-Client's sind dagegen vollwertige Rechner. Sie haben einen normalen Rechner als Basis. Berechnungen werden zumeist hier ausgeführt. Der Server dient vorrangig zum zentralen-Verwalten und Speichern von Daten.

Da Client und Server praktisch nur agierende Software-Produkte sind, können Client und Server auch auf dem gleichen Rechner laufen. Eine räumliche Trennung von Client und Server sind gut möglich und meist auch die Grundlage, aber es ist keine notwendige Voraussetzung für eine praktische Kommunikation.



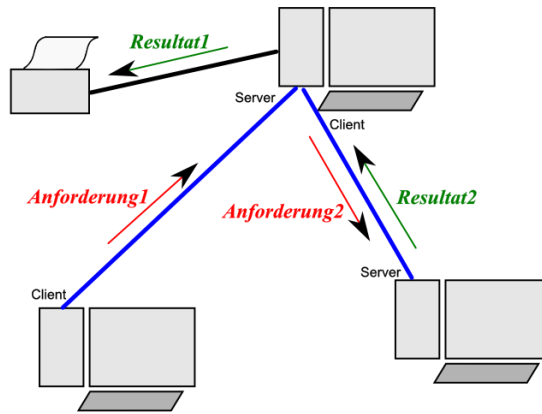
Die Client-Server-Kommunikation ist der vorrangige Typ im Internet. Man spricht auch vom Client-Server-Prinzip (Client-Server-Paradigma).

Als Gegen-Konzept wird oft das Peer-to-peer-System genannt. Hierbei sind die Geräte sachlich gleichberechtigt. Wenn man allerdings genau hinsieht, dann ist auch in einem Peer-to-peer-Netzwerk auch die Rollen-Verteilung in Client und Server realisiert. Allerdings sind die Geräte hier immer beides – Client und Server. Nur im Augenblick sind mehr mit einer Rolle beschäftigt.

Damit die Kommunikation in dynamischen Netzen mit verschiedensten Client's und Servern vonstatten gehen kann, müssen auf allen Geräten eindeutig ansteuerbare Kommunikations-Endpunkte eingerichtet sein. Man spricht auch von Service Access Points oder Sockets.

Diese sind Betriebssystem-übergreifend festgelegt.

Ein Socket ist eine IP-Adresse und eine definierte Port-Nummer. An diese Port-Nummer ist dann die zugehörige Anwender-Software angebunden.



## Lokale Netzwerke (LAN / WLAN)

praktisch privat durch spezielle Adressen

Broadcast-Prinzip (Rundfunk- / Rundruf-Prinzip)

Übertragungsmedium wird von allen Nutzer gemeinsam genutzt

Daten werden praktisch alle Nutzer gesendet

jeder Empfänger nutzt nur die Pakete, die für einen selbst gedacht ist

alle anderen ignorieren das Paket (Achtung!: hier besteht immer Lausch-Gefahr!)

Ziel-Adresse ist im Header des Daten-Paketes eingebaut

Adressen müssen innerhalb eines Netzes eindeutig sein

Adressen sind Werte-Kombinationen in bestimmten Formaten

local Host: 127.0.0.1

lokaler Gateway / Router (häufig): 192.168.0.1

nennt man IP-Adressen (Internet Protokoll-Adressen)

bestehen aus Präfix, der das Netzwerk charakterisiert

und Suffix, der den individuellen Host angibt

Sender:

(W)LAN-Karte erhält Daten von einem Programm

packt die Daten in ein oder mehrere definierte Pakete

setzt in Pakete die Ziel-Adresse und die eigene Absender-Adresse ein

ergänzt noch technische Informationen

---

gibt dann das Paket auf dem Medium (Kabel oder Funk) aus  
alles direkt auf dem Netzwerk-Adapter (Netzwerk-Karte) ohne Beteiligung der CPU usw.

**Individual-Adressen:**

Adresse eines Host's (gemeint eines Netzwerk-Adapter's)

**Multicast-Adressen**

gemeinsame Adresse der Host's eines Netzwerkes bzw. eines Bereichs daraus (Gruppen-Adresse)

**Broadcast-Adressen**

gemeinsame Adresse aller Host's eines Netzwerkes (immer alle Host's angesprochen)

z.B. für technische Zwecke

Broadcast-Domains sind alle Rechner, die in einem Netzwerk ein per Broadcast versendetes Signal empfangen können

**MAC-Adressen**

Media Access Control Address

Hardware-Adresse des Netzwerk-Adapter's

praktisch weltweit einmalig

vom Hersteller vergeben (teilweise aber änderbar, muss aber innerhalb des Netzes eindeutig bleiben)

12-stellige Hexadezimal-Zahl (eigentlich 6 Byte Dualzahl), Doppelpunkt-getrennt

Datenpakete bestehen immer aus Header (Kopf-Daten) und Nutzdaten

Prä- ambel	Ziel- Adresse	Quell- Adresse	Typ	Nutzdaten	...	CRC
---------------	------------------	-------------------	-----	-----------	-----	-----

**Kopplung von LAN's**

***Kopplungs-Bauteile in Netzwerken***

- **optisches Modem** für längere Entfernungen (verbinden Netzwerke über anderes Medium)
- **Repeater** Zwischen-Verstärker (innerhalb eines Netzes)
- **Hub** koppeln von Netzwerken bzw. Host's mit der gleichen Technologie  
Daten werden an alle Teilnehmer weitergereicht
- **Bridge** Koppeln von Netzwerken mit verschiedenen Technologien
- **Switch** koppeln von Netzwerken bzw. Host's mit der gleichen Technologie  
Daten werden nur an den Empfänger weitergereicht
- **Router**

# Ethernet

wichtigste LAN-Technologie

entwickelt von METCALFE und BOGGS Anfang der 1970er Jahre

Kabel-gebundenes System

Ethernet-Kabel ist das Medium dieser Technologie (Ether genannt)

Übertragungs- und Kollisions-Vermeidungs-Technologie → CSMA/CD (Carrier Sense Multi Access / Collision Dedection))

Festlegung der Daten-Pakete → Ethernet-Frame's

Prä- ambel	Ziel- Adresse	Quell- Adresse	Typ	Nutzdaten	...	CRC
8 Byte	6 Byte	6 Byte	2 Byte	45 – 1'500 Byte		4 Byte

Präambel-Bytes sind 10101010-Dualzahlen → wecken die Empfänger auf und synchronisieren die Host's

letztes Präambel-Byte ist 10101011

Verbindungs-loser (connectionsless) Service → keine (sichere) Punkt-zu-Punkt-Verbindung unzuverlässiger (non-reliable) Dienst, da es selbst bei unvollkommenen Übertragungen keine Bestätigung (Acknowledgement) oder Fehlermeldung an den Sender gibt Fehler-Korrektur nur über höhere Netzwerk-Schichten / übergeordnete Programme möglich

## CSMA/CD

alle Empfänger lauschen

immer nur ein Paket zu einem Zeitpunkt übertragbar

wenn das Medium frei ist, dann sendet der Sender ein Paket praktisch an alle

ist Medium besetzt, dann wird eine kurze (zufällige) Zeit (Back-off-Time) gewartet und das Senden erneut versucht

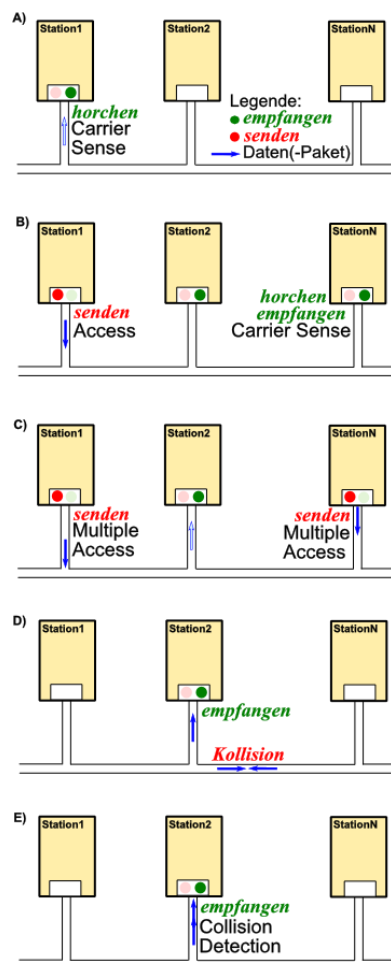
bei zufälligen Kollisionen (unsaubere Signale, Fehler in den Daten, ...) wird Übertragung abgebrochen (JAM-Signal)

zu eine Collision-Domain gehören alle die Host's, zwischen denen es beim gleichzeitigen Senden zu einer Kollision kommen kann

Ethernet-Segment sind alle die Rechner, die über ein Ethernet-Kabel (einschließlich Hub's und Switches) verbunden sind bildet somit eine Collisions-Domäne

Repeater verbinden zwei (gleich-adressierte) Ethernet-Segmente zu einer Collision Domain

Bridge verbinden zwei (gleich-adressierte) Ethernet-Segmente ohne eine gemeinsame Collision Domain



---

Switches verbinden zwei (gleich-adressierte) Ethernet-Segmente / Host's ohne eine gemeinsame Collision Domain

Switch entscheidet über die Weiterleitung eines Paketes auf einem ausgewähltem Port (oder dem Verwerfen)

Router verbinden zwei unterschiedliche adressierte Ethernet-Segmente ohne eine gemeinsame Collision Domain

entscheidet über die Weiterleitung eines Paketes in ein anderes LAN (oder dem Verwerfen)

### **Ethernet-Medien**

- **Koaxial-Kabel** Ring- oder Bus-Topologie  
bis 10 Mbit/s  
ab 1980
- **Twisted Pair** Stern-Topologie  
10 / 100 / 1000 Mbit/s  
100 Mbit-Version heißt Fast-Ethernet; ab 1995  
ab 1999 Gigabit-Ethernet: 1000 Mbit/s = 1 Gbit/s  
ab 2001 gibt es für MAN und GAN das 10 Gbit-Ethernet  
ab 2015 dann auch 2,5 oder 5,0 Gbit/s z.B für PoE (Power over Ethernet) (günstigere Hardware und Verkabelung)
- **Glasfaser** ab

weiterhin Unterscheidung nach:  
Anzahl der anschließbaren Host's  
Bandbreiten / Topologie (s.a. oben)

### **Charakteristika:**

- mehrere Hundert Rechner in einem LAN
- Ausdehnung bis rund 1 km

### **Vorteile:**

- relativ hoher Datendurchsatz
- geringe Verzögerung in der Daten-Übertragung (da keine Speicher oder Transport-Logik gebraucht werden)
- einfache Algorithmen
- faires System
- hohe Zuverlässigkeit
- sehr stabil unter hoher Last
- 90 % der Kapazität werden (max.) praktisch genutzt

### **Nachteile:**

- unsicher, da alle Nutzer ständig lauschen und fremde Pakete auch nutzen können
- keine Korrektheits-Kontrolle oder Fehler-Korrektur
-

---

## WLAN – lokale Funk-Netzwerke

Wireless Local Area Network  
Kabellose Netzwerke

mögliche Mobilität ist entscheidender Vorteil

Ursprung war ALOHAnet, das die Hawaiianischen Haupt-Inseln verbunden hat, ab 1971 ein Zentral-Rechner (Access-Point), mehrere Clients; Verbindung 9'600 bit/s = bps)  
praktisch eine Stern-Topologie

Standard IEEE 802.11  
802 charakterisiert Netzwerke  
11 die Funk-Standard's  
angehängter Buchstabe bestimmt der Version

### **WLAN-Standard's**

- **a**      54 Mbit/s  
            5 GHz
  
- **b**      11 Mbit/s  
            2,4 GHz  
            Verschlüsselungs-Verfahren WEP (Wired Equivalent Privacy)
  
- **i**      Verbesserung des veralteteten Verschlüsselungs-Verfahren WEB auf WPA und WPA2
  
- **g**
- **n**
- **ac**      Ziel: mehr Geschwindigkeit
  
- **ax**      WiFi 6; neuester Standard (Nachfolger von ac)  
            Ziel: mehr Host's  
            ab 2018

moderne WLAN-Router sind Dualband-Geräte für 2,4 und 5 GHz

Frequenz-Band	2,4 GHz	5 GHz
<b>Merkmale</b>		seltener genutzt
<b>Vorteile</b>	vor allem auch von älteren Geräten unterstützt höhere Reichweite	mehr überlappungs-freie Kanäle
<b>Nachteile</b>		



---

### **Hidden-Station-Problem**

durch Überlappung der Funk-Reichweiten

z.B. mehrere WLAN-Access-Point's in einem Netzwerk

entfernte Host's haben keine Informationen zum Funk-Betrieb an anderen Access-Point's

mehrere Access-Point's empfangen Signale von einem Host (ev. leicht Zeit-versetzt)

hier jetzt CSMA/CA-Verfahren

CSMA wie bei Ethernet

CA Collision Avoidance ()

- potentieller Sender sendet RTS-Signal (Request To Send)
- verfügbarer Empfänger sendet CTS-Signal (Clear To Send)
- aktivierter Sender sendet nun DS-Signal (Data Send) und dann ein Daten-Paket
- aktivierter Empfänger sendet ACK-Signal als Bestätigung des Empfang's (Acknowledgement)
- alle Funk-Geräte, die RTS/CTS hören, warten (eine zufällige Zeit, bis dann wieder ein RTS gesendet wird)
- dadurch können Kollisionen nur noch im Bereich der RTS- und CTS-Signale passieren (nicht mehr aber beim Daten-Senden)

Exposed-Station-Problem

trotz überlappender Funk-Bereiche liegen Access-Point's soweit auseinander, dass sie untereinander keinen direkten Funk-Verkehr aufmachen können (es wird eine Zwischen-Station gebraucht)

MACAW-Verfahren (Multiple Access with Collision Avoidance for Wireless)

- entferntes Gerät sendet RTS
- eine freie Zwischenstation (ohne benutzte Verbindungen in seinem Bereich) sendet nach einer kurzen Pause mit einem RRTS (Request for Request To Send) (Pause notwendig, damit direkte Empfänger vorher mit CTS antworten können(, dann wird Zwischenstation nicht benötigt))
- weiter, wie beim CSMA/CA

### **Sicherheit in Funk-Netzen**

potenziell unsicher, da jeder Funk mitempfangen kann

Parkplatz-Attacke möglich

passives Mithören der Funk-Verbindungen (Mitschneiden des Daten-Verkehrs) praktisch spurlos möglich

ev. durch Cracken oder statistischen Analysen auch Mithören von verschlüsselten Verbindungen

aktives Erforschen / Lesen / Schreiben im angeschlossenen Netzwerk

---

## **Verschlüsselungs-Verfahren**

- **WEP**  
**Wired Equivalent Privacy** IEEE 802.11b  
RC4-basiert  
durch Mitschneiden des Funkverkehrs lassen sich Teile des Schlüssels erschließen
- **WPA**
- **WPA2**  
u.a. z.B. AES-basiert
- **WPA3**  
zukünftiger Standard  
deutlich verbesserte Authentifizierung und Kryptographie  
Perfect Forward Secrecy

## **Aufgaben:**

- 1.
2. *In einem vom Internet-Provider oder selbst gekauften "WLAN-Router", wie z.B. eine "Fritz!box" sind verschiedene Netzwerk-Geräte-Typen integriert. Welche sind das und welche Funktionen erfüllen sie?*
- 3.

## **Netze für größere Entfernungen – WAN's**

Wide Area Network

Broadcast hat hier wegen der Signal-Laufzeiten keine Bedeutung mehr  
erste WAN's entstehen durch Zusammenschluß von LAN's untereinander und / oder Host's  
es entsteht meist praktisch ein virtuelles, offenes Netzwerk – ein kleines Internet

Daten werden in Pakete verpackt, ev. (im Header) als zusammengehörig markiert  
auf den (ev. auch unterschiedlichen) zur Verfügung stehenden Verbindungen verteilt weitergeleitet

Verfahren ist Paket-Vermittlung (Packet Switching)

die passenden Geräte heißen Router (bringen Packet auf die (richtige) Tour (Route))

Router haben meist schon eine spezialisierte Hardware, aber auch spezielle Software (oft Linux-basiert)

Entscheidung wird aufgrund der Adress-Informationen (Präfix) getätigt

Pakete für fremde (nicht für das eigene) Netzwerke werden weitergeleitet, die für das eigene Netzwerk werden verworfen (weil sie finden intern einen Abnehmer /anderen Host))

Weiterleitung über das Next-Hop-Forwarding

der Router kennt eine nächste Weiter-Vermittlungs-Stelle (Next-Hop)

dafür gibt es die Routing-Tabelle

in der Tabelle sind weitere Netzwerke verzeichnet

Routing-Algorithmus

Auswerten der Ziel-Adresse

---

passenden Eintrag in Routing-Tabelle suchen  
Weiterleiten des Paket's über die verzeichnete Verbindung

## **Routing im WAN**

Routing ist das Bestimmen des optimalen Wegs eines Pakets vom Sender zum Empfänger  
Auswertung der Ziel-Adresse aus dem Daten-Paket  
Router wählt ein angeschlossenes Netzwerk für das Daten-Paket aus  
Leitet das Paket in dieses Netz unter Verwendung der gebrauchten Technologie  
praktisch Quellen-unabhängig (dem Router ist egal, wo das Paket herkommt, ihn interessiert nur das Ziel)  
benutzt wird Optimalitäts-Prinzip, das besagt, dass die optimale Route nur vom Ziel abhängt (zurückgelegter Weg oder Quelle ist unwichtig)  
dadurch Next-Hop-Algorithmus das optimale Verfahren

im Router ist das gesamte Netzwerk (als Graph) gespeichert  
im Graph sind Knoten die Host's und die Kanten die Verbindungen  
praktisch Tabelle von angeschlossenen Host's  
jeder Router hat eine eigene – für ihn spezifische – Tabelle mit Host-Adresse (Präfix-Adress-Teil) und dem nächsten notwendigen Zwischen-Punkt (Next-Hop)  
für alle möglichen Ziele muss ein passender Eintrag in der Routing-Tabelle vorhanden sein (→ universelles Routing) und es muss die optimale (kürzeste / billigste / mit den wenigsten weiteren Zwischenpunkten / größte Übertragungsrate) Verbindung (Route) eingespeichert sein

Verbindet man die optimalen Verbindungen aller angeschlossenen Host's H1, H2, ..., Hn mit dem Ziel Z, dann entsteht ein Wurzel-Baum (Sink Tree). Dieser ist nicht eindeutig bestimmt. Sink Tree hat eine natürliche Metrik, d.h. es existiert eine Aussage über die Anzahl weiterer Sprünge (HOP's) zum Ziel (Host). Im Sink Tree sind Schleifen ausgeschlossen, was dafür sorgt, dass nach endlich vielen Sprüngen das Ziel auch erreicht wird.

## **Routing-Algorithmen**

konstruieren einen Sink Tree für ihre eigene Netzwerk-Position und den angeschlossenen / erreichbaren Netzwerken / Host's

zentrale Routing-Algorithmen

zentrale Instanz (zentraler Host) berechnet ein Routing-Tabelle  
relativ rechen-intensiv, da Dynamik im Netz auch Dynamik in der Tabelle bedeutet, außerdem muss die (zentral-ermittelte) Tabelle immer wieder und überhaupt verteilt werden  
Problem bei Fehlern → Fehler hier hat Konsequenzen für das gesamte System (Single Point of Failure)  
auch immer Angriffs-Punkt für Attacken → bedeutet effektive Attacke

dezentrale Routing-Algorithmen

praktisch genutzt

jeder Router berechnet seine eigene Routing-Tabelle  
dazu werden die nächstgelegenen Router nach ihren angeschlossenen Netzwerken befragt  
bedeutet aber auch relativ lange Anpassungs-Phase  
dazu gehören z.B. auch:

- isoliertes Routing
- Distanzvektor-Verfahren

- 
- Link-State-Routing
  - ...

## Cloud's und Cloud-Computing

Cloud (Wolke) als Symbol für das Internet  
Nutzung von virtuellen Maschinen ((leistungsschwache) Endgeräte sind nur noch Ein- und Ausgabe-Geräte)

typische / bekannte Anwendungen:

- google drive
- one drive
- dropbox
- iCloud
- Bdrive
- ...
- microsoft Azure
- amazon cloud
- telekom cloud
- ...
- HiDrive
- teamdrive

ermöglicht:

- Zugang zu Daten von jedem beliebigen Internet-Standort
- kollaboratives Arbeiten
- hohe Verfügbarkeit von Daten (wikipedia, google, ...)
- online-Gaming-Apps
- online Apps (Nutz-Programme)
- anspruchsvolle Programme auf leistungsschwachen Geräten

mögliche Probleme:

- viele (ev. sensible) Daten werden im Internet verarbeitet
- Konkurs von Cloud-Betreibern
- Weiter-Verkauf von Daten durch Cloud-Betreiber?
- versteckte Kosten
- Verfügbarkeit (Was passiert, wenn man mal keine Internet-Verbindung hat?)

## 8.3. Internet – das Netz der Netze

notwendige Geräte:

- Endgeräte (Rechner, Laptop's, Tablet's, Smartphone's, Lampen / Rolläden / Steckdosen / ..., Fernseher, Spielekonsolen)
- LAN / WLAN
- Router / Modem

Account beim Internet-Service-Provider (ISP)

Charakteristika des Internet's

- virtuelles Netzwerk
- offenes / unfertiges Netzwerk
- Multi-Protokoll-Netzwerk

### 8.3.1. Kopplung der Vielzahl von Netzwerken - Internetworking

Netze unterscheiden sich stark – heterogenes Netz

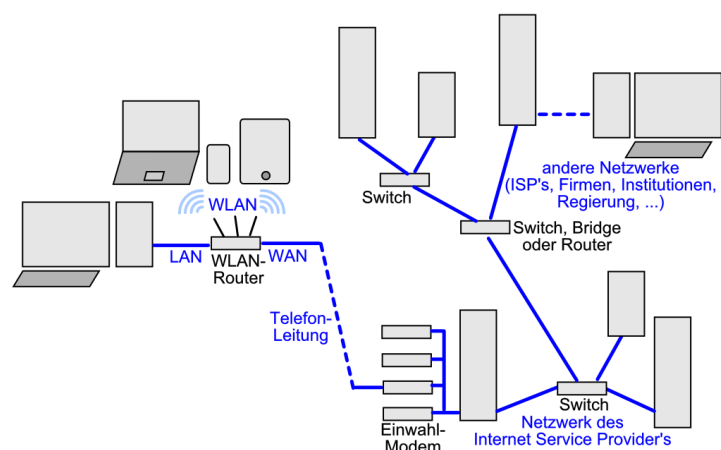
jeweils begrenzt für ihren Einsatz-Zweck bestimmt, ausgelegt und optimiert  
einfacher Zusammenschluß geht nicht, wegen inkompatibler Adressierungen und Protokolle

Internetworking ist das Konzept zum Zusammenschluß heterogener Netze zu einem einheitlichen Kommunikations-System (Internet)

es wird quasi ein virtuelles (Gesamt-)Netzwerk erzeugt

Netzwerke sind offen, können und sollen nicht durch zentrale Instanzen kontrolliert und gesteuert werden

Netzwerk muss sich selbst organisieren



**Probleme:**

- Adressierung über verschiedene Adressierungs-Systeme hinweg
- Paket-Formate und -Größen (z.B. größere / kleinere Pakete)
- Ziel-Findung über verschiedene Netzwerke hinweg, wobei von anderen Netzwerken praktisch nichts bekannt ist
- unterschiedliche Arten von physikalischen Signalen
- Berechnung von Routen (über weitgehend unbekannte Zwischen-Stationen)
- einheitliche / vergleichbare Fehler-Korrekturen (Fehler-Erkennung, Übertragungs-Fehler)
- Überlast auf Verbindungen
- ...

## Internet-Protokoll-Stapel

verschiedene gefühlte Aufgaben-Ebenen werden in Schichten / Protokoll-Stapel organisiert  
→ TCP/IP ()

Standard's / Protokolle / (verschiedene) Software frei zugänglich

innerhalb eines Netzwerkes eindeutige Adressen, aber nicht über das gesamte Internet  
es können z.B. die Router (in heimischen Netzwerken) immer 192.168.0.1 als Adresse haben

hätten sie diese Adresse auch nach außen, dann wären sie nicht unterscheidbar – es gäbe eine IP-Adress-Konflikt

Internet-Adressierung ist hierarchisch strukturiert

Umsetzung von Next-Hop-Verfahren

Zerlegung der Daten in definierte Pakete

unabhängige Übertragung der Pakete im Netz (→ durch unterschiedliche Paket-Wege kann sich die Empfangs-Reihenfolge ändern)

einzelne Pakete können fehlerhaft übertragen werden oder verloren gehen

Router verbindet 2 Netzwerke

Routing-Verfahren

- Sender erzeugt IP-Paket
- Netzwerk-Karte erzeugt z.B. Ethernet-Paket für ein LAN
- Router entnimmt IP-Paket aus dem Ethernet-Paket, prüft die Empfänger-Adresse und erzeugt ev. ein Übertragungs-Paket für das 2. Netz (z.B. WLAN-Paket)
- Router sendet WLAN-Paket ins 2. Netz

Internet-Protokoll-Stapel

notwendig sind:

- Adress-Protokolle
- Kommunikations-Protokolle
- Daten-Paket-Standard's

Zerlegung der Gesamt-Aufgabe auf bestimmte Teil-Aufgaben

da unterschiedliche Komponenten unterschiedliche Zugriffs-Bereiche haben, wird ein Schichten- bzw. Layer-Modell benutzt

zwischen den Schichten gibt es definierte / standardisierte Schnittstellen

jede Schicht kann und darf nur mit den angrenzenden Schichten kommunizieren (Schichten dürfen aber in einer Umsetzung zusammengefasst werden (ev. Verzicht auf Kompatibilität zu Lösungen mit "normaler" Schicht-Struktur))

Schichten sind immer Lösungen für spezielle Probleme / Aufgaben (von denen andere Schichten gar keine Kenntnisse haben müssen)

z.B. muss ein eMail-Programm nicht wissen, ob die eMail's über ein Ethernet oder ein WLAN übertragen werden. Diese Aufgaben-Ebene wird an passende Schicht delegiert

Im Normal-Fall müssen

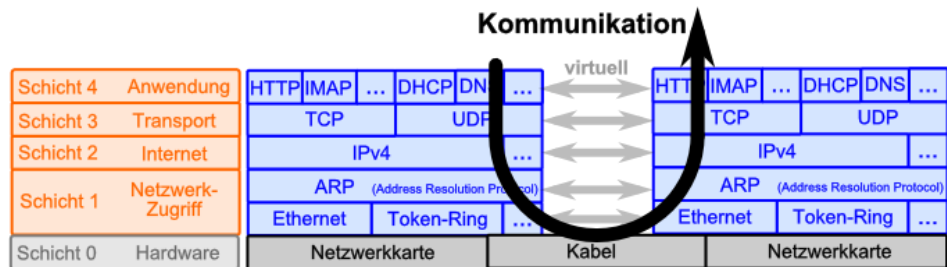
Sender und Empfänger mit dem gleichen Schichten-Modell arbeiten

Viele Schicht-Modelle zueinander kompatibel.



Gegenüberstellung mehrerer Schicht-Modelle  
(links: OSI-ISO-Modell, DoD-Modell, TCP/IP-Modell)

Schicht-Folge  
beim Sender  
wird beim Empfänger  
umgekehrt



praktisch kommunizieren die zusammengehörenden Schichten bei Sender und Empfänger –  
zumindestens virtuell - miteinander

Ebene	Schicht od. Layer	Aufgabe
4	Application-Layer	Verarbeiten der Daten und Bereitstellung der zu übertragenden Daten für den Transport ins Netz bzw. die Rück-Übersetzung in die Datenstruktur des Anwender-Programm's z.B.: FTP, HTTP, SMTP, ...
3	Transport-Layer	universeller Transport-Dienst (beinhaltet z.B.: Auf- und Abbau von Verbindungen; Übertragungs-Garantien, Korrektur von Paketfolgen, ...)
2	Internet-Layer	Transport-Dienst für die Daten-Pakete Verarbeitung der Adressierungen und dann Routing
1	Link-Layer	logische Interpretation / Umsetzung der Daten in physikalische Signale und umgekehrt (MAC-Bereich: Zugriff auf's Übertragungsmedium; LLC-Bereich: logische Verbindungen und Fehler- u. Fluss-Kontrolle) Bit-Ströme in Paket gepackt und umgekehrt ev. Ergänzung von Informationen zur Fehler-Korrektur
(0)	(Hardware)	Datentransport über physikalisches Medium

LLC ... Logical Link Control

Nachteil des Schichten-Modell:

- längere (weil nicht optimalste) Verarbeitung der Daten
- zusätzliche Daten der einzelnen Schichten (→ Overhead)
- häufiges Ein- und Auspacken

Routing-Kriterien:

- Datendurchsatz
- Kosten
- Lastverteilung
- Sicherheit
- ...

---

## Vermittlungs-Systeme / Zwischen-Systeme

- **Repeater** arbeiten auf unterster / physikalischer Schicht (Hardware-Layer)  
reine Signal-Verstärkung → Ausdehnung der Reichweite  
innerhalb einer Collision domain  
unsichtbar für höhere Schichten  
keine eigene Intelligenz
- **Hub**
- **Bridge** verbinden 2 LAN-Segmente  
gehören zur Link-Schicht  
ermöglichen LAN-Erweiterung (z.B. über Spezifizierungs-Grenzen hinweg)  
intelligente Paket-Vermittlung (Verkehrs-Management)  
trennt die Collision domains voneinander  
in jedem Segment unabhängiger Daten-Verkehr möglich (jeweils bis Voll-Last)
- **Switch** verbinden 2 oder mehr LAN-Segmente / Host's  
gehören zur Link-Schicht  
ermöglichen LAN-Erweiterung (z.B. über Spezifizierungs-Grenzen hinweg)  
intelligente Paket-Vermittlung (Verkehrs-Management)  
trennt die Collision domains voneinander  
in jedem Segment unabhängiger Daten-Verkehr möglich (jeweils bis Voll-Last)
- **Router** verbinden unterschiedliche (eigenständige / autarke) Netze miteinander  
höhere Intelligenz notwendig  
gehören zur Internet-Schicht  
kennt alle angeschlossenen Netze  
brauchen neben Prozessor und Speicher (ROM+RAM) noch Netzwerk-Karten / -Anschlüsse der beiden anzuschließenden Netzwerke  
Übertragen von Daten-Paketen und Adress-Schemata des einen Netzes in die Pakete / Adressen des anderen Netzes (quasi doppelte Intelligenz)
- **Gateway** eigentlich: Application Level Gateway  
verbindet auf der Applikations-Ebene (Application-Layer)  
dienen der Kommunikation von verteilten Anwendungen oder verteilten Komponenten eine Anwendung auf verschiedenen Geräten  
übersetzen Anwendungs-Protokolle ineinander

sorgen immer für die Paket-Vermittlung und die Adress-Übertragung zwischen den gekoppelten Netzwerken

### 5-4-3-Repeater-Regel

Ein Netz das nur Repeater verwendet, darf aus maximal 5 Segmenten bestehen. Somit sind nur 4 Repeater möglich. Nur 3 der Segmente dürfen Endgeräte enthalten. Die restlichen Verbindungen müssen Punkt-zu-Punkt-Verbindungen (Inter Repeater Links (IRL)) zwischen Repeatern sein.



---

## Zugänge zum Internet

typisch notwendig ISP (Internet Service Provider (z.B.: Telekom, Vodafone, unity.media, Kabel-Deutschland, ...))

bestimmen die Abrechnungs-Modelle (Flatrate, nach Daten-Menge oder Nutzungs-Zeit)  
oft mit (Festnetz-)Telefon und / oder Kabelfernsehen gekoppelt

### **Zugangs-Medien für Normal-Nutzer**

#### **stationär**

- **analoges Telefon** notwendig Modem (Umsetzung von digitalen Daten auf Ton-Signal und umgekehrt)  
Kupfer-Leitungen (Klingel-Draht ohne Abschirmung(en))  
max. 55 Kbit/s
- **ISDN** zusätzliche digital-ausgelegte Übertragungs-Bänder auf der analogen Telefon-Leitung (mit ersten Abschirmungen und BUS-Techniken)  
ab 1992  
64 oder gebündelt 128 Kbit/s  
Kapazität durch die benutzten Frequenz-Bänder nahe am akustischen Telefon
- **DSL** weitere Frequenz-Bänder und Übertragungs-Technologien auf der Basis von Kupfer- / Draht-Telefon-Leitungen  
ab 2000  
üblich im Standard bis 2 Mbit/s  
begrenzend sind die Kabellänge → immer mehr Interferenzen und Störungen  
neue Versionen ADSL (ab 2000)  
schaffen heute (mit ADSL2, ADSL2+, VDSL, VDSL2) deutlich höhere Übertragungs-Raten (bis 100 Mbit/s) durch dichte Abstände zum nächsten Umsetz-Punkt  
derzeit VDSL-Vectoring
- **Kabel**  
**Coaxial-Kabel**  
**Fernseh-Kabel** ab 1980 und vor 1997 einfache Kabel-Anschlüsse (ausschließlich für den Empfang von Fernsehen und Radio); nur eine Übertragungs-Richtung (unidirektional)  
ab 1997 dann erweiterte Kabel-Systeme mit Rückkanälen (DOCSIS-Standard)  
Coaxial-Kabel lassen deutlich höhere Übertragungs-Raten (bis 200 Mbit/s) und größere Kabellängen (bis rund 150 Kilometer)  
technische Grenze derzeit: 10'000 Mbit/s
- **Glasfaser-Kabel** neue Leitungen notwendig; anspruchsvollere Technik
- **Richtfunk**
- **Satellit** Signal-Empfang über Satellit  
je nach Technik Sende-Technik auch über Satellit oder über einfache DSL-Leitung (→ SkyDSL mit 50 Mbit/s (Download) ab 1999)

---

hohe Latenz-Zeiten (500 bis 700 ms)  
neue Satelliten-System (Starlink (Musk)) mit rund 60 weltweit verfügbaren Satelliten; seit 2019; da niedrige Umlaufbahnen sind kurze Latenzzeiten im System)

•

## **mobil**

- **WLAN** privat oder öffentlich / geschäftlich (Free WiFi, ...)
- **Mobil-Funk** GSM seit 1990 (A-, B- und C-Netz) = 2G(eneration) mit max. 9,6 Kbit/s (Deutschland: GSM900: Uplink: 890 – 915 MHz; Downlink: 935 – 960 MHz)  
2005 EDGE mit 473,6 Kbit/s  
dann 2007 3G mit HSDPA und 42 Mbit/s  
ab 2010 4G (LTE) mit 1'200 Mbit/s (Deutschland: 800 MHz; 1,8 + 2 + 2,6 GHz) + LTE-Advanced  
5G ab 2019 mit 10 Gbit/s  
neben Handy / Smartphon auch Laptop's usw. möglich (mit zusätzlichen USB-Adapter (für SIM-Karte))

•

### Mobil-Funk

nutzt zwei unterschiedliche Frequenz-Bänder für Up- und Down-Link (wegen der unterschiedlichen angebotenen Daten-Durchsätze)

## 8.3.2. Protokolle der Vermittlungsschicht - das Internet-Protokoll IP

Aufgaben des IP-Protokoll's:

Fragmentierung und Defragmentierung von Daten(-Paketen)

da alle Formate individuelle Größe der Last-Daten haben und Daten-Menge i.A. die MTU (Maximum Transfer Unit = maximale Datenpaketgröße) überschreitet, müssen Daten in kleinere Pakete geteilt werden (Fragmentierung)

Pakete werden nummeriert (als Fragment-Nummer)

Beim Ziel müssen die empfangenen Daten-Fragmente wieder zu einem Daten-Bestand zusammengesetzt werden (Defragmentierung)

Bestimmung der optimalen Route

da Internet sehr groß ist (x-Mrd. Adressen) kann der einzelne Router nicht mehr das gesamte Netz kennen und somit auch nicht mehr die optimale Route bestimmen

Ziel ist natürlich die Pakete über möglichst wenige Zwischenstationen und ohne Schleifen zum Empfänger zu transportieren

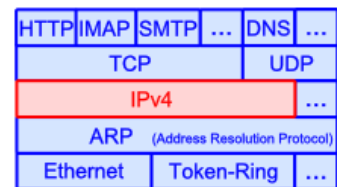
zentrales Kommunikations-Protokoll für die Internet-Schicht

für Paket-Vermittlung verantwortlich

notwendig ist ein hier einheitliches Adressierungs-Schema

Auffinden optimaler Routen

Erkennen von Übertragungs-Fehlern (z.B. über Prüfsummen)



	IPv4	IPv6
<b>Beispiele</b>	141.89.255.101 192.168.0.1 127.0.0.1	2001:638:807:204::8d59:e17e
<b>Merkmale</b>	32 Bit-Folge 4x 8 Bit, notiert als dezimale Zahl, mit einem Punkt getrennt	128 Bit-Folge als Hexadezimal-Zahlen in 4er Gruppen notiert; Doppelpunkt getrennt
<b>Gemeinsamkeiten</b>		
<b>Unterschiede</b>		
<b>Vorteile</b>		
<b>Nachteile</b>		

Fragmentierung

dann notwendig, wenn aktuelles Paket in seiner umgepackten Version im neuen Netz die maximale Paket-Größe (MTU) überschreiten würde

ursprüngliches Paket wird in solche Fragmente geteilt, die im neuen Netz transportiert werden können und dann ein Header mit entsprechenden Fragment-Nummern ergänzt (Fragmentierung)

## Fragmentierungs-Arten

- transparent** Router baut aus zu großen Daten(Paketen) kleine Pakete und sendet diese zum nächsten Router etc.  
 der nächste Router setzt das ursprüngliche Paket wieder zusammen und sendet es im nächsten Netz weiter (ev. unter dem erneuten Umbau in kleinere Pakete für dieses Netz)  
 Nachteile: ständiges Zerlegen und Zusammensetzen; gleiche Route für alle Detail-Pakete notwendig
- nicht-transparent** einmal verkleinerte Pakete werden auf dem weiteren Weg nicht wieder vervollständigt (zum großen Daten-Paket) sondern in der verkleinerten Form weiterversendet  
 Nachteile: (sehr) viele kleine Pakete mit vielen Kopfdaten

Empfänger muss im Normalfall immer die Daten-Pakete zusammensetzen (in Netzwerk-Karte auf Schicht )

u.U. auch aus unterschiedlichen Fragmentierungs-Systemen, weil Daten unterschiedliche Wege gegangen sein können

Problem bei der Fragmentierung kann z.B. sein, dass einzelne Fragmente beschädigt sind. Da nur dieses zur wiederholten Übertragungs beim Sender angefordert wird, ist dies für den Sender problematisch, da er die Fragmentierungen, die auf dem Weg vorgenommen wurde, gar nicht kennt / kennen kann.

Um nicht auch extrem kleine Daten-Paket-Größen in Protokollen möglich wird, wurde eine atomare (minimale) Fragment-Größe festgelegt. Kleiner dürfen Fragmente nicht werden.

4e	00	0	D	A	T	E	N	P	A	K	E	T
			Daten des Fragment's									
			Ende-Bit (Steuerungs-Bit)									
			Fragment-Nummer des 1. Fragment's im Daten-Paket									
			Paket-Nummer									

die atomare Fragmentgröße ist hier mit 1 Byte angenommen (Länge des Datenteil's hier 10 Byte).

### 1. Fragmentierung

Zerlegung in Fragmente mit einer Datengröße von 5 Byte

4e	00	0	D	A	T	E	N	4e	05	1	P	A	K	E	T
----	----	---	---	---	---	---	---	----	----	---	---	---	---	---	---

es entstehen 2 Pakete, deren gemeinsame Kennung die Paket-Nummer 4e ist  
 das erste (linke) Paket ist das 0. Fragment, hat aber Nachfolger, da das Ende-Bit nicht gesetzt ist

Das 2. Fragment hat 5 Byte Vorgänger-Daten und ist das letzte, da hier das Ende-Bit gesetzt ist.

### 2. Fragmentierung

Nur für 2. Fragment der 1. Fragmentierung (weil z.B. in anderem Netz übertragen mit kleiner MTU!). Das andere Fragment wird in unveränderter Form in einem anderen Netz übertragen.

4e	05	0	P	A	K
----	----	---	---	---	---

4e	08	1	E	T
----	----	---	---	---

4e	00	0	D	A	T	E	N
----	----	---	---	---	---	---	---

Die Paket-Nummer ist einheitlich bei 4e geblieben. Das 2. Fragment wurde in Pakete der Länge 3 Byte zerlegt, wobei vom 2. Fragment dieser Fragmentierung nur 2 Byte Daten genutzt werden. Die neuen (Unter-)Fragmente sind jetzt durch eine neue Fragment-Nummerierung abgestimmt. Das 1. (Unter-)Fragment behält seine Fragment-Nummer, aber das Ende-Bit wird auf 0 gesetzt, weil es jetzt nicht mehr das letzte Fragment ist. Das zweite (Unter-)Fragment erhält eine passende Fragment-Nummer (Nummer + Datenlänge des letzten Fragment's). Da es das letzte Fragment ist, wird das Ende-Bit gesetzt.

Die Übertragung der Fragmente kann auf unterschiedlichen Wegen und in unterschiedlichen Geschwindigkeiten passieren. Beim Empfänger müssen die Fragmente wieder zum ursprünglichen Paket / Datensatz zusammengesetzt werden.

4e	08	1	E	T
----	----	---	---	---

4e	00	0	D	A	T	E	N
----	----	---	---	---	---	---	---

4e	05	0	P	A	K
----	----	---	---	---	---

**Defragmentierung** erfolgt dann durch logisches Verbinden der Fragmente:

Betrachtet werden nur die Pakete mit der gleichen Paket-Nummer – in unserem Fall die 4e. Das erste Paket wird anhand der ersten Fragment-Nummer – also 00 identifiziert.

4e	00	0	D	A	T	E	N										
----	----	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--

Da das Ende-Bit nicht gesetzt ist, muss es weitere Fragmente geben. Das Paket mit der kleinsten Fragment-Nummer ist nun 05. Das passt auch zur bisher zusammengesetzten Paket-Länge (erste 5 Zeichen aus erstem Fragment).

Die Daten aus dem Fragment können also sofort angehängt werden:

4e	00	0	D	A	T	E	N	P	A	K							
----	----	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--

Das Ende-Bit im 2. Fragment war wieder nicht gesetzt, also muss mindestens noch ein Fragment existieren. Also wird naxh dem Fragment mit der kleinsten Fragment-Nummer gesucht. Das 3. Fragment mit der Nummer 08 passt wieder direkt an die bisher defragmentierten Daten:

4e	00	1	D	A	T	E	N	P	A	K	E	T					
----	----	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--

Beim 3. Fragment ist das Ende-Bit gesetzt, womit die Defragmentierung beendet ist.

### 8.3.2.1. Internet-Protokoll Version 4 (IPv4)

stammt aus dem Jahr

gehört zur Internet-Schicht (Layer 2)

derzeit sehr stabil und immer noch gerne benutzt, weil einfach

Problem ist begrenzter Adressraum, der derzeit praktisch ausgeschöpft ist

Router zwischen zwei technologisch unterschiedlichen Netzen

erhält ein Paket des Sende-Netzes in dessen Format (z.B.: Ethernet-Paket), in dem das IP-Paket eingebettet (als Nutzlast) ist

entpackt das IP-Paket

verpackt das IP-Paket (als Nutzlast) in das Paket-Format des Ziel-Netzes (aus der Sicht des Router's) und sendet es in das Ziel-Netz (auf der Basis der Routing-Tabelle im Router)

das IP-Paket hat keine Information oder Zugriffs-Möglichkeit auf die umgebenden Paket-Formate und dessen Protokolle, es ist nur Daten-Last

Adressierungs-Schema (nur für IP-Pakete gültig)

Aufbau eines IP-Paket's:

max. Länge 65'535 Byte

geteilt in Header (Kopfdaten) und Data (Nutzdaten)

Header 20 – 60 Byte nach Paket-Art und -Version

Datagramm (eng.: Datagram)

Byte	+1		+2	+3	+4
0	IP-Version	IHL	Dienst-Art	Paket-Länge	
4	Identifikation		Flag's	Fragment-Zähler	
8	TTL		Protokoll	Header-Kontrollsumme	
12	Quell-Adresse				
16	Quell-Adresse				
20	Ziel-Adresse				
24	Ziel-Adresse				
28	Nutz-Daten				
...					

IHL .. Länge des Header's (min. 5 und max. 25 von 32-Bit-Worten)

Dienst-Arten:

Priorität (0 .. 7), Delay (), Throughput (), Reliability ()

Identifikation (aus IP-ID genannt) ist die Nummerierung der Pakete (z.B. durch den Router)

TTL .. Time to Live (Lebensdauer) wird bei jedem Hop um 1 verkleinert (→ dekrementiert), wenn 0 erreicht ist, dann wird Paket verworfen

Protokoll ist die Angabe des übergeordneten Transport-Protokoll's (z.B. )

Header-Checksum (Header-Kontrollsumme), wird bei jedem Hop neu berechnet (da sich ja z.B. die TTL jedesmal ändert!)

---

### **Optionen im Header**

- **strict source routing** Vorgabe des zu benutzenden Routing-Pfad's
- **loose source routing** unbedingt zu nutzende Zwischen-Stationen
- **record route** Protokollierung der Zwischen-Stationen-Adressen
- **time stamp** Router ergänzt seine Adresse und einen Zeitstempel

### **Routing bei IPv4**

größere Netzwerke, die z.B. zu einem Internet-Service Provider gehören oder regionale Einheiten (z.B. Netze eines Staates) werden als autonome Systeme (AS) betrachtet. Innerhalb der AS wird als Routing-Protokoll ein Interior Gateway Protocol (IGP) benutzt. Das könnte z.B. OSPF (Open Shortest Path First) sein.

Beim OSPF wird ein Link-State-Routing durchgeführt, das den DIJKSTRA-Algorithmus benutzt. Dadurch kann man sich schnell an dynamische Veränderungen des Netzwerkes anpassen, hierarchisch Routen, sowohl in LAN's, wie auch in WAN's arbeiten, kann Hersteller-unabhängig agieren und unterschiedliche Metriken (z.B. kürzester Weg; billigste Route, wenigste Hop's, ...) als Bewertungs-Grundlage benutzen.

### ***Nachrichten-Typen zwischen Routern (IGP)***

- **Hello** Wer sind meine Nachbarn?
- **Link State Update** Link-State-Aktualisierung an Nachbarn (Status, Metrik, Kosten)
- **Link State Ack** Bestätigung der Link-State-Aktualisierung
- **Database Description** Link-State-Pakete mit aktuellen Informationen des Senders
- **Link State Request** Link-State-Anforderung an Nachbarn zum Ermitteln der aktuellsten Verbindungs-Daten

Ziel des Routing innerhalb eines Autonomen Systems ist eine möglichst effektive Paket-Zustellung.

Zwischen autonomen Systemen benutzt man dagegen ein Exterior Gateway Protocol (EGP), wie z.B. BGP (Border Gateway Protocol).

Die Router, die an den Schnittstellen zwischen den verschiedenen Autonomen Systemen angeordnet sind, sind für beide Protokolle ausgelegt. Man spricht dann von Multi-Protokoll-Routern.

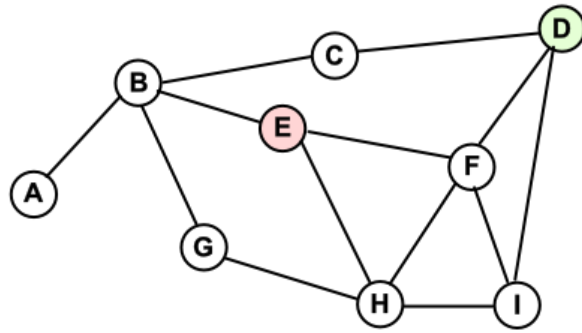
Bei EGP müssen auch politische, wirtschaftliche und sicherheitstechnische Aspekte beachtet werden, da i.A. auch Staatsgrenzen überschritten werden.

Als Transport-Protokoll wird TCP benutzt. Dadurch wird die Verbindung zuverlässig und es werden Details zum benutzten Netz verschleiert.

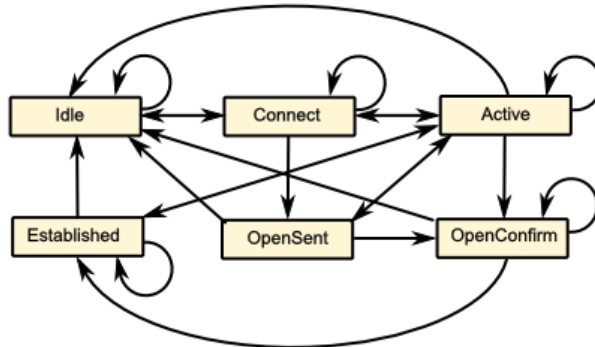
Distanzvektor-Routing, wobei sowohl die Distanz, als auch der zugehörige Pfad gespeichert und verwendet wird. Das BGP teilt den Nachbar-Systemen z.B. auch die verwendeten Pfade mit.

Die Zwischen-Station E bekommt von den direkten Nachbarn B, F und H die folgenden Routen zum Ziel D geliefert:

- B – C – D
- F – D
- H – I – D



(→ **Sprachen und Automaten**)



Zustands-Graph für das Border Gateway Protocol

Adress-Umfang  
 direkt ansprechbar: ca. 4 Mrd. Host's ( $2^{32} =$ )

IANA verteilt die Adressen auf die kontinentalen Verteiler (Regional Internet Registries (RIR))



## Exkurs: DIJKSTRA-Algorithmus

spricht: deik.stra  
entwickelt von edgar DIJKSTRA ()

Weg-Finde-Algorithmus in einem Graphen G mit gewichteten Kanten

Ziel ist das Finden aller kürzesten Wege von einem Start-Knoten A zu allen anderen Knoten

Gegeben ist ein beliebiger Graph mit Informationen über die Kanten (z.B. Länge, Leitungs-Anzahl, Leitungs-Durchmesser, Daten-Kapazität, ...). Die genaue Interpretation des Wertes interessiert den Algorithmus nicht. Es müssen nur addierbare Werte sein.

### Vorgehensweise:

- zuweisen von Distanz und Vorgänger zu jedem Knoten
- erstellen zweier leeren Knoten-Listen ("besucht" und "gefunden")
- initialisieren der Distanz im Start-Knoten A mit **0** und aller anderen Knoten mit **unendlich** (oder Wert über Maximalwert (z.B. Summe aller Kanten))
- SOLANGE es noch einen unbearbeitete Ziel-Knoten gibt
  - wählen eines Knoten mit minimaler Distanz
  - diesen in der Liste "gefunden" speichern
  - für alle Nachbar-Knoten, für die es noch keine kürzeste Distanz gibt
    - berechnen des Kantengewicht's und der aktuellen Distanz
    - ist der Knoten noch nicht in der "besucht"-Liste, dann hinzufügen
    - ansonsten prüfen, ob der gerade berechnete Wert kleiner ist, als die in der "Besucht"-Liste gespeicherte Distanz
    - aktualisieren der Distanz
    - aktuellen Knoten als Vorgänger setzen

Beispiel:

→ recht gute Erklärung bei "50 Jahre Internet" → Video: internetworking2019-3-e1-pip.mp4

### Aufgaben:

1. *Realisieren Sie den DIJKSTRA-Algorithmus in einer einfachen Programmiersprache (z.B. Python)! Verwenden Sie obiges Beispiel! Lassen Sie sich immer bestimmte / informative Zwischenwerte anzeigen*
2. *Verändern Sie den Algorithmus so, dass für die einzelnen Ziel-Knoten auch eine Liste der Zwischen-Knoten auf dem kürzesten Weg gespeichert wird!*
- 3.

### 8.3.2.2. Internet-Protokoll Version 6 (IPv6)

aktuelle Version

aktuell erst ein Viertel des Internet auf diese Version umgestellt

notwendig wegen des begrenztes Adress-Umfangs von IPv4

Fehler bei der ursprünglichen Domain-Festlegungen

Routing wird dadurch komplizierter, weil Unternetze definiert werden müssen

große Tabellen; derzeit trotz konzeptioneller Einfachheit sehr komplex geworden

Bedarf heute je eine Adresse für jeden Sensor / Aktor / ... (IoT); Mobile-Geräte

keine Möglichkeiten für Festlegungen von Dienst-Qualitäten (Quality of Service) z.B. für ruckelfreie Video's; Life Audio- / Video-Übertragungen

keine Unterstützung von Gruppenarbeiten (kollaboratives Arbeiten)

schon 1994 über Nachfolger nachgedacht (IETF .. Internet Engineering Task Force))

bewährte Konzepte wurden beibehalten:

verbindungsloser Paket-Dienst

eigenständige / unabhängige Paket-Übertragung

Anzahl der maximalen Hop's

neues Konzept:

128 bit Adress-Raum (acht 16 bit Gruppen in hexadezimaler Notation; Doppelpunkt als Trenner; Nullen-Kompression)

ergibt  $2^{128} = 3,403 \cdot 10^{38}$  Adressen ( $340,3$  TTT Adressen =  $340,3 \cdot 10^{12} \cdot 10^{12} \cdot 10^{12}$ ) entspricht  $10^{23}$  mögliche Adressen pro  $m^2$  Erdoberfläche

führende Nullen dürfen weggelassen werden

zusammenfassen und weglassen einer längsten Kette aus Nullen, nur die angrenzenden Doppel-Punkte werden mitgeschrieben

#### Aufgaben:

1.

2. **Bestimmen Sie die kürzeste Notierung der folgenden Voll-Adressen im IPv6-Format!**

a)

b)

d)

e)

g)

h)

i) 11011111 11010010 10010011 11010101 11100101 01000101 11110100 10001011 ...  
... 11100011 00000000 00000000 00000000 00000000 01001111 11001010 00001111

3.

Teil	Präfix (Side-ID)	Subnet- ID	Interface-ID
IPv6-Adresse	<b>2001:453a:01d3:</b>	<b>0007:</b>	<b>0000:0000:026b:f38d</b>
Länge	6 Byte	2 Byte	8 Byte

Der Präfix beschreibt den Typ der Adresse () oder ist die Netzwerk-Adresse einer Firma, eines ISP usw. usf.

Mit der Subnet-ID können im Firmen-Netzwerk einzelne Unter-Netzwerke definiert werden.

Mit der Interface-ID wird der Host adressiert. Diese Adresse kann z.B. aus der MAC-Adresse berechnet werden.

Ein Subnetz-Maske, wie bei IPv4 ist nicht notwendig, da die Teilung von Netzwerk- und Interface-Teil immer bei 64 bit (8 Byte) erfolgt.

verschiedene Adress-Typen für  
Unicast, Multicast und Cluster

neue Header  
mehrere Formate zugelassen

Basis-Header ist obligatorisch

diverse Optionen möglich, vor allem für Authentifikation, Verschlüsselung, Fragmentierung,  
...

verbesserte Unterstützung von Audio und Video  
besonders für Echtzeit-Übertragungen festgelegte Übertragungs-Pfade möglich

Protokoll ist nun erweiterbar

neue Funktionalitäten (z.B. Mobile IPv6; Neighbor Discovery Protocol (ICMPv6, was ARP ersetzt)

Auto-Konfiguration der Host's (Stateless DHCP)

Unterstützung von Multihoming und Renumeration

### 8.3.3. Protokolle der Transport-Schicht

müssen auf Verbindungs-losen Vermittlungs-Dienst (IP) aufsetzen

Schicht 4	Anwendung	HTTP	IMAP	...	DHCP	DNS	...	
Schicht 3	Transport	TCP		UDP				
Schicht 2	Internet	IPv4				...		
Schicht 1	Netzwerk-Zugriff	ARP (Address Resolution Protocol)						
		Ethernet		Token-Ring				...

#### **Protokolle auf der Transport-Schicht**

- **TCP**  
**Transmission Control Protocol**
- **UDP**  
**User Datagram Protocol**
- **ISO/T1 ... ISO/T4**
- **NetBIOS**
- **RTP** für Multimedia-Daten  
**Real Time Protocol** Life-Streaming
- **SNA**  
**System Network Architecture**
-

---

### **8.3.3.1. TCP – Transmission Control Protocol**

ist der Erfolgsgarant des Internet

besonders Leistungs-fähig

- ermöglicht auf der Grundlage des unzuverlässigen IP-Paket-Vermittlungs-Dienst einen zuverlässigen, gesicherten / garantierten Transport-Dienst
- korrigiert "Fehler" des IP
- voll duplex (bidirektional)
- veränderlich / weiterentwickelbar
- bei Paket-Verlusten oder fehlerhaften Übertragungen werden Pakete wiederholt angefordert → Retransmission (adaptive Neuübertragung)
- Überlast-Kontrolle (Congestion Control)

#### **Merkmale / Charakteristika:**

- Verbindungs-orientierter Dienst (Ende-zu-Ende-Übertragung)
- immer nur Verbindung von 2 End-Systemen (exakt nur zwischen 2 Anwendungen auf den beiden End-Systemen)
- nur die beiden End-Systeme kommunizieren auf dieser Ebene (untergeordnete Layer werden versklavt)
- virtuelle (Verbindungs-orientierte) Verbindung auf der Basis einer in Wirklichkeit Verbindungs-losen Kommunikation
- den Endsystemen wird eine stabile Verbindung vorgegaukelt
- Nachrichten (TCP-Pakete) werden mit Sequenz-Nummern versehen
- Empfänger (TCP-Ebene) prüft Sequenzen und sortiert ev die Paket
- TCP muss nur von den End-Systemen (Sender und Empfänger gekannt werden; Router verstehen / brauchen nur IP)
- TCP nutzt also die IP-Paket-Erstellung und –Übertragung; TCP-Pakete sind in IP-Paketen gekapselt
- für IP ist das TCP-Paket eine "unleserliche" Datenlast, wie jede andere Datenstruktur
- vor dem eigentlichen Datenaustausch muss erst eine virtuelle Verbindung aufgebaut werden
  - beim ersten Verbindungs-Aufbau werden initiale Sequenz-Nummern ausgetauscht (und bestätigt)
  -
- am Ende muss Verbindung geschlossen werden

#### **Nachteile**

- kein Multi- oder Broadcast

z.B. genutzt für:

-

## Aufbau eines TCP-Paket's

	4	8	12	16	20	24	28	32
0	source port				destination port			
32	sequence number							
64	acknowledgement number							
96	data offset	reserved	flags			window		
128	checksum				urgent pointer			
...	options 0 .. n 32-bit-words							
	data							

### flags

	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN
<b>Name</b>	Congestion Window Reduced	ECN-Echo	Urgent	Acknowledgement	Push	Reset	Synchronise	Finish
<b>Verfahren</b>	Explicit Congestion Notification (ECN)							
<b>Umschreibung / Bedeutung</b>	Datenrate ist passend	Netz überlastet Datenrate reduzieren	dringend	Quittierung	Einschalten des Überganges des Eingangs- und Ausgangs-Puffers	Abbruch / Zurücksetzen der Verbindung	Initiierung einer Verbindung	Beenden / Freigabe der Verbindung

durch das Einkapseln des TCP-Paketes in ein IP-Paket entsteht ein sogenannter Pseudo-Header  
 praktisch sind die IP-Information vorgelagert  
 allerdings wird die Checksumme im TCP-Paket benutzt und an das gesamte IP-Paket angepasst

#### bei IPv4

	4	8	12	16	20	24	28	32
0	<b>source adress</b>							
32	<b>destination adress</b>							
64	<b>0000 0000</b>		<b>0000 0110 (=TCP)</b>			<b>TCP-length</b>		
96	source port				destination port			
128	sequence number							
160	acknowledgement number							
192	data offset	reserved	flags			window		
224	<b>checksum</b>				urgent pointer			
...	options <i>0 .. n 32-bit-words</i>							
	data							

#### bei IPv6

	4	8	12	16	20	24	28	32
0	<b>source adress</b>							
32								
64								
96								
128	<b>destination adress</b>							
160								
192								
224								
256	<b>TCP-length</b>							
288	0000 0000		0000 0000		0000 0000		next header	
320	source port				destination port			
352	sequence number							
384	acknowledgement number							
416	data offset	reserved	flags			window		
448	<b>checksum</b>				urgent pointer			
...	options <i>0 .. n 32-bit-words</i>							
	data							

## Verbindungs-Aufbau für eine TCP-Nachrichten-Übertragung

da das darunterliegende IP keine sichere Ende-zu-Ende-Verbindung ermöglicht, muss diese durch TCP nach-realisiert werden

praktisch ein virtuelle Verbindungs-orientierte Kommunikation

TCP-Nachrichten werden mit Sequenz-Nummern von jedem End-Gerät durchnummeriert

### 3-Wege-Handshake

Vorrangig dient der Verbindungs-Aufbau zur Vereinbarung der initialen Sequenz-Nummern und damit der Absicherung der Verbindung.

Der Client – der ja irgend eine Anfrage an eine Server hat – betrachten wir hier m al als Sender. Dieser erzeugt eine zufällige Sequenz-Nummer für seine Verbindung.

Der Sender schickt nun ein TCP-Paket an den Empfänger. Im Paket ist das SYN-Flag gesetzt.

Der Empfänger prüft den im Paket angegeben Ziel-Port. Ist dieser geöffnet (also eine passende Applikation verfügbar), dann wird das SYN-Flag auf Null gesetzt und dafür das ACK auf Eins.

Damit wird der Empfang der Verbindungs-Anforderung (sowie das erhaltene TCP-Paket) quittiert. Im Antwort-Paket wird nun die Sequenz-Nummer des Senders um Eins erhöht und eine eigene Sequenz-Nummer erzeugt.

Das Antwort-Paket geht nun an den Empfänger zurück.

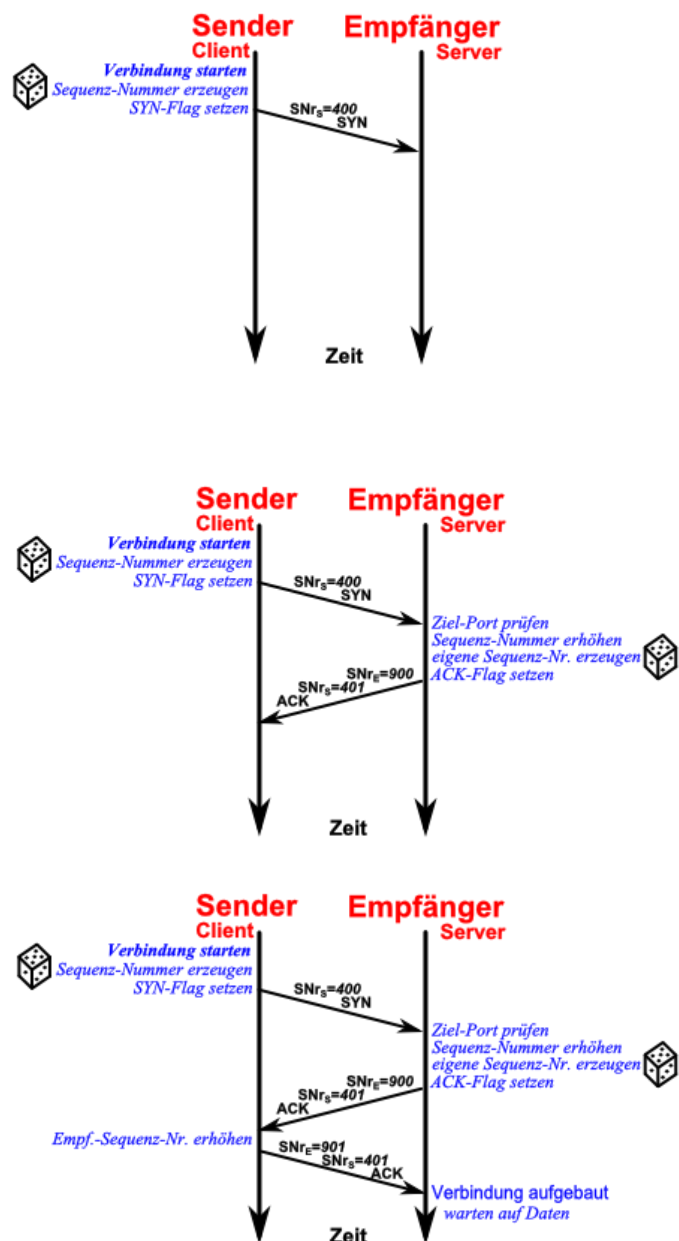
Ist der Ziel-Port nicht geöffnet, dann wird ein TCP-Paket mit gesetztem RST-Flag.

Der ursprüngliche Sender nimmt nun das Antwort-Paket entgegen, prüft, ob die Sequenz-Nummer erhöht wurde. Die Sequenz-Nummer, die ihm der Empfänger übermittelt hat, wird nun wiederum vom Sender erhöht.

Im 3. Schritt wird nun ein Quittierungs-Paket mit den quasi bestätigten Sequenz-Nummern und dem ACK-Flag an den Empfänger zurückgeschickt.

Damit ist die Verbindung vereinbart und der Empfänger wartet nun auf Daten (die eigentliche Anfrage) vom Sender.

Wegen der drei notwendigen Einzelschritte nennt man das Verfahren auch 3-Way-Handshake (umgangssprachlich: 3-Wege-Handschütteln)



notwendig ist ein Abbau der Verbindung nach dem Ende der Kommunikation (→ )

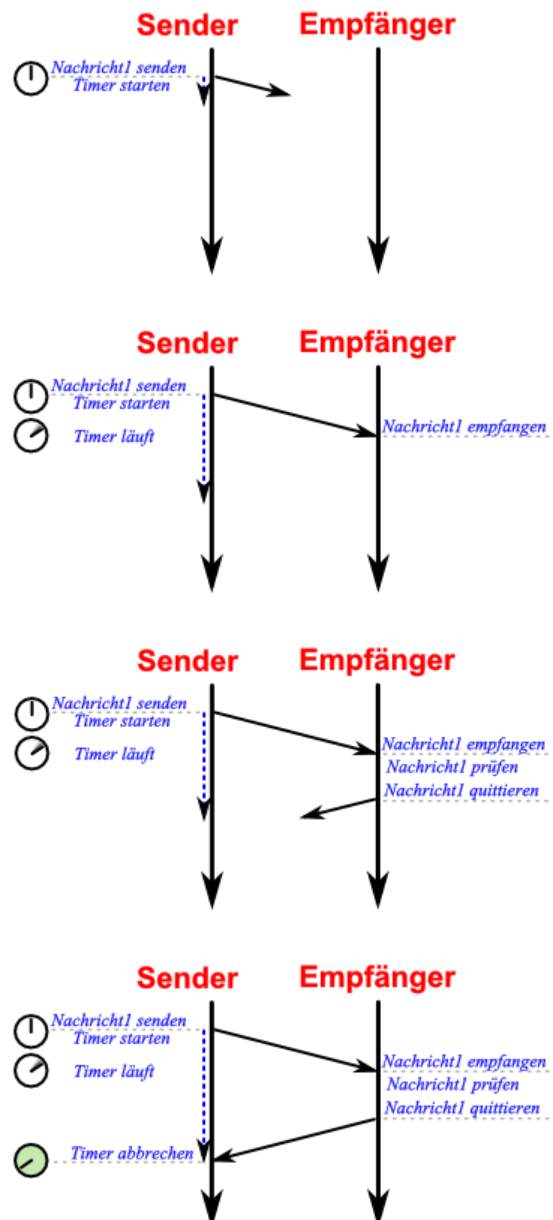
## Ablauf / Verlaufs-Protokoll einer TCP-Nachrichten-Übertragung

Der TCP-Senderschickt ein Daten-Paket los. Das Paket wird auf der IP-Schicht gekapselt und über das IP-Netzwerk zum Empfänger geroutet. Beim TCP-Sender wird gleichzeitig ein Timer gestartet. Er beinhaltet die maximale Wartezeit auf eine Quittierung.

Irgendwann hat der Empfänger die Nachricht (hier Nr. 1) erhalten. Er prüft die Nachricht über die Kontrollsumme und die vorher vereinbarte Session-Nummer.

Ist alles in Ordnung, wird eine Quittierung (Acknowledgement) vom Empfänger an der sender zurückgeschickt.

Läuft alles gut, dann kommt die Quittierung vor dem Ablauf des Timer's an. Damit ist die Übertragung von Nachricht1 aus der Sicht des Sender's abgeschlossen und er kann zur Übertragung der nächsten Nachricht übergehen.



Timer ist Bremsschuh der Daten-Übertragung. Ist er zu kurz eingestellt und die Zeit recht nicht für eine normale Quittierung, dann wird ständig das gleiche Daten-Paket abgeschickt. Wählt man die Timer-Zeit zu lange, dann können weniger Daten übertragen werden, weil u.U. mehr fach (sehr lange) gewartet werden musste. Weiterhin kommt es zu einem Datenstau zwischen der Anwendung und der TCP-Schicht. Eine hier positionierte Warteschlange könnte überlaufen und damit vielleicht Daten verloren gehen

Timer muss auch für unterschiedliche Netze unterschiedlich sein. In einem lokalen LAN kann die Zeit recht klein ausfallen. Bei großen Netzwerken (z.B. einem WAN) sollte man die Timer länger einstellen.



## Retransmission (Neuübertragung von fehlenden oder fehlerhaften Datenpaketen)

Der Start der nächsten Nachricht2 erfolgt nach dem gleichen Schema.

Nun kann das Daten-Paket entweder vor dem Empfang oder während der Quittierung verloren gehen. Weiterhin können die Daten irgendwie verändert worden sein. In dem Fall würde die Kontrollsumme nicht mehr stimmen. Es kommt – wodurch auch immer bedingt – keine Quittierung zum Sender zurück.

Irgendwann läuft der Timer ab.

empfangt der Sender keine Bestätigung, dann sendet er das Paket einfach noch einmal. Sender wartet eine bestimmte Zeit auf die Quittierung

dazu verwendet er den RTT-Timer

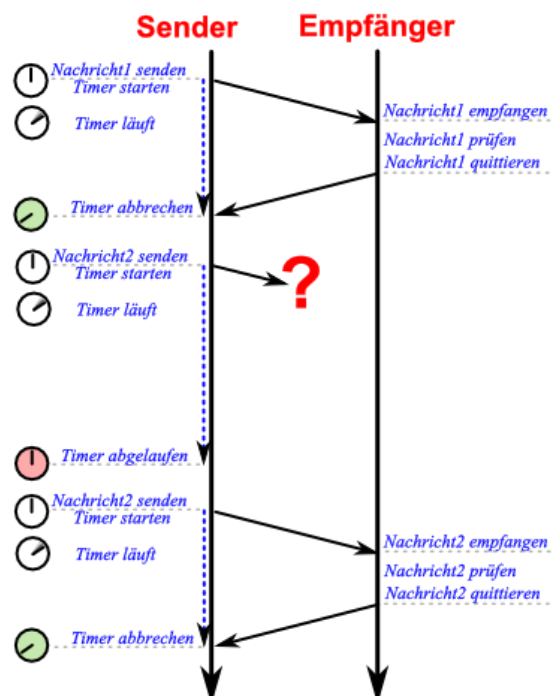
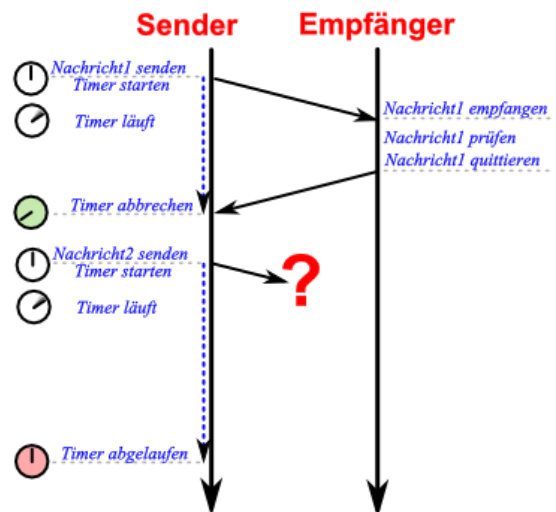
ist die Zeit abgelaufen, ohne dass eine Bestätigung erfolgt, dann wird Paket noch einmal versendet

Nachricht2 wird nun übertragen, als wenn vorher nichts gewesen wäre. Läuft alles gut, kommt diesmal die Quittierung vorm Timer-Ablauf

alle Handlungen werden vom TCP erledigt  
die Anwendungs-Programme erzeugen nur die Daten für die TCP-Schicht

Die TCP-Schicht überlässt den weiteren Transport einer IP-Version

Was mit den TCP-Paketen weiter passiert entzieht sich dem Zugriff durch die Schicht-Struktur.



Da eine einheitliche Timer-Festlegung nicht allen Situationen gerecht werden kann, wird hier die adaptive Retransmission benutzt.

Dazu wird ständig die Zeit für einen Paket-Umlauf (Versand bis Quittierung) gemessen. Diese Zeitspanne wird Round-Trip-Time (RTT) genannt und nach bestimmten Verfahren (gleitendes Mittel) die Smoothed Round-Trip-Time berechnet. Sie steht als Kennwert für die aktuelle Netzwerk-Situation.

## Aufgaben:

1.

2. Berechnen Sie das jeweils gültige gleitende Mittel über drei Daten-Punkte!

Messwert	2	3	5	4	2	1	5	5	3	1	2	3	2	3	4	5	3	2
gleit. Mittel																		

3. Stellen Sie die Messwerte (als Punkte) und das gleitende Mittel (als Kurve / Gerade) in einem Diagramm dar!

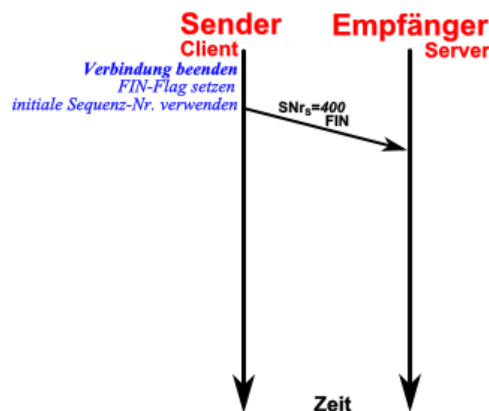
4. Interpretieren Sie den Graphen! Warum wählt man das gleitende Mittel für die Berechnung der Timer-Zeit (Smoothed-Trip-Time) für den TCP-Sender und nicht z.B. das genauere arithmetrische Mittel?

## Abbau einer TCP-Nachrichten-Übertragung

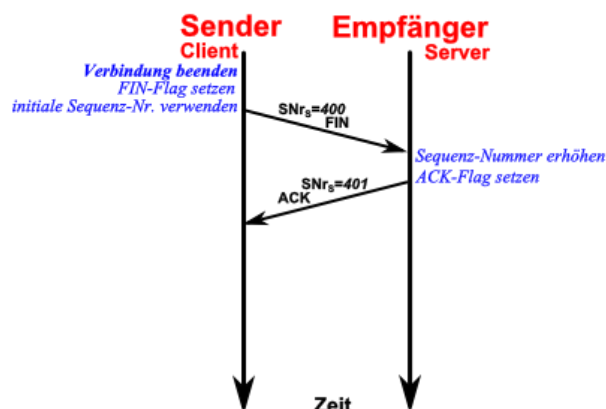
auch TCP-Teardown

Praktisch wird wieder der 3-Wege-Handshake benutzt und je nach Implementierung um eine weitere Quittierung erweitert (praktisch → 4-Wege-Handshake).

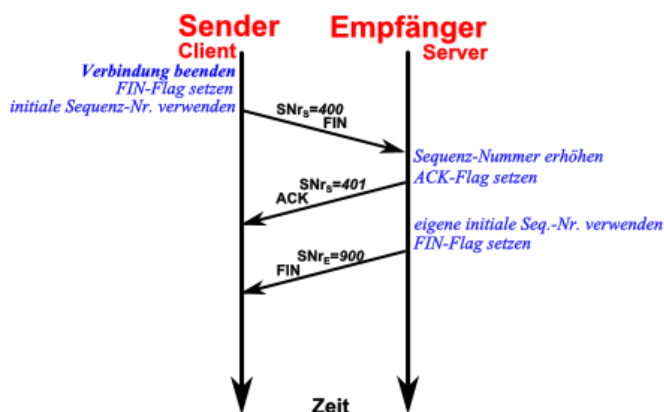
Zum Beenden der Verbindung wird wieder auf die initiale Sequenz-Nummer zurückgegriffen und ein Paket mit dieser und dem gesetzten FIN-Flag versendet.



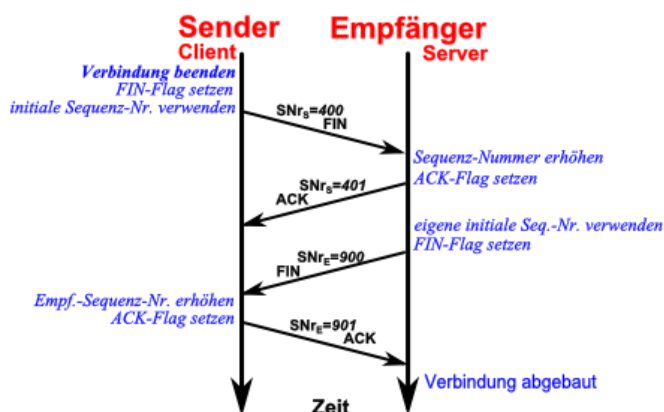
Der Empfänger quittiert dem Empfang des FIN-Signal's mit dem Zurückschicken eines Paket's mit der inkrementierten Sequenz-Nummer und dem üblichen ACK-Flag.



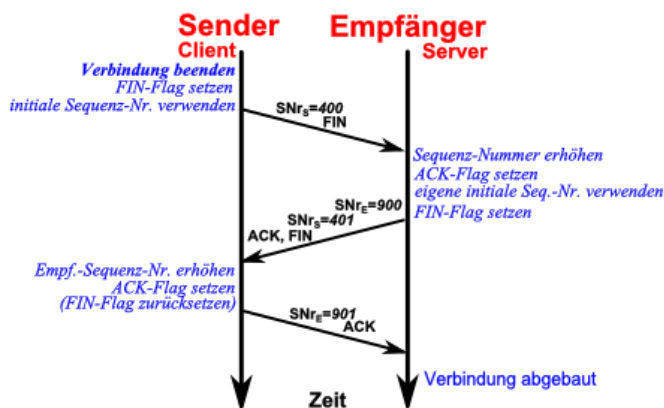
Danach wird vom Empfänger ein Paket gebaut, in dem dieser seine initiale Sequenz-Nummer und sein FIN-Signal verwendet. Dieses Paket wird dann an den ursprünglichen Sender verschickt.



Der Sender bestätigt wiederum dieses Paket durch Erhöhen der Sequenz-Nummer und dem Setzen des ACK-Flag's. Nach Erhalt dieses Paket's beim ursprünglichen Empfänger ist die Verbindung beendet. Alle Pakete mit der inkrementierten Sequenz-Nummer des ursprünglichen Senders werden nun ignoriert.



In einigen Implementationen ist die Quittierung des Sender-FIN's mit dem Aussenden des eigenen FIN-Signal's verbunden. Dadurch verkürzt sich das Verfahren auf ein 3-Wege-Handshake.



## Fluß- und Überlast-Kontrolle im TCP

### Sliding Window Protocol (Schiebefenster-Protokoll)

Fenster sind hier Gruppen von Übertragungseigenschaften (Parameter) bei unterschiedlichen Bedingungen sollten sich die Parameter von selbst anpassen die Zuweisung und die Quittierung eines Parameter-Fenster's sollte wieder voneinander entkoppelt sein

Bei großen Daten-Paketen (z.B. 2'000 Byte) kann der Empfänger mit der Menge überfordert sein. Z.B. könnte sein Eingangspuffer nur 1'000 Byte groß sein. Der Sender schickt nun ein

---

Paket mit 750 Byte. Dieses wird vom Empfänger auch bestätigt und gleichzeitig mitgeteilt, dass nur noch für 250 Byte Platz im Eingangs-Puffer ist. Diese 250 Byte sind nun die neue Fenster-Größe.

Da der Sender nun die Größe übermittelt bekommen hat, versendet er das nächste Paket nur mit 250 Byte.

Damit ist er Eingangs-Puffer des Empfängers voll und er sendet ACK und eine Fenster-Größe von 0 Byte zurück, wenn der Puffer immer noch belegt ist.

Das ist nun das Signal für den Sender nicht weiter Daten zu übertragen. Er geht in den Warte-Modus.

Ist beim Empfänger nun (durch die Ziel-Applikation) der TCP-Eingangs-Puffer geleert worden – ev. auch nur teilweise – dann überträgt der Empfänger die neue freie Fenster-Größe (freier Puffer-Platz) an den Sender. Dieser kann nun wieder Daten in der freien Größe übertragen.

Das Verfahren wird entsprechend fortgesetzt, bis alle Daten übertragen sind.

Ein Problem kann entstehen, wenn die Fenster-Größe einen sehr kleinen Wert hat. Der Sender würde dann ja nur kleine Pakete nachschicken, was sehr ineffektiv wäre (→ **Silly Window Syndrom**).

In der Praxis ist es nun so, dass der Empfänger nicht jede beliebige, gerade frei gewordene Puffer-Größe meldet, sondern wartet, bis der Platz mindestens der Hälfte der maximalen Fenster-Größe entspricht. Auch der Sender nutzt nicht die gesamte zur Verfügung stehende Fenster-Größe aus, sondern sendet etwas kleinere Daten-Pakete, damit immer eine kleine Reserve im Puffer ist und es nicht Puffer-Überläufen kommt. Diese könnten die Integrität des Empfänger-Betriebssystem gefährden.

## **Congestion Control (Überlast-Steuerung)**

Da das TCP nur an den End-System aktiv ist, kann das Protokoll von sich aus keine Probleme bei der Daten-Übertragung (z.B. übervolle Leitungen) erkennen. Weiterhin können Pakete ja völlig unterschiedliche Wege durch nutzt gehen. Jeder der Wege kann unterschiedlich stark belegt sein.

TCP nutzt nun ein indirektes Verfahren, um die Übertragung zu beurteilen. Da die empfangenen Pakete bestätigt werden, kann der Sender diese ins Verhältnis zu nicht bestätigten – also irgendwie verlorengegangenen – Paketen setzen. Wird das Verhältnis ungünstig, dann wird das ECE-Flag übertragen und das andere End-Gerät kann seine Fenstergröße reduzieren. Stimmt die Fenster-Größe, wird das CWR-Flag gesendet. Dies besagt, dass die Fenster-Größe ok ist.

### Slow-Start-Algorithmus

Bei diesem Verfahren beginnt der Sender mit relativ kleinen Paketen. In den nachfolgenden Übertragungen erhöht (verdoppelt) er die Fenster-Größe solange, bis zu viele Pakete nicht mehr bestätigt werden. Dann wird eine etwas geringere Fenster-Größe eingestellt und benutzt.

### Congestion-Avoidance-Algorithmus

Werden zu viele gesendete Pakete nicht bestätigt, dann wird die Sende-Rate einfach verringert, bis wieder akzeptable Werte erreicht werden.

### NAGLE-Algorithmus

Das vorrangige Ziel ist es bei diesem Verfahren, den Overhead möglichst gering zu halten. Sind die Daten-Anteile im Vergleich zum Header ungünstig, dann wird die Quittierung hinausgezögert. Es werden dann z.B. auch eigene – zurück zuschickende Daten – gesammelt und mit in die Übertragung einbezogen.

Für jedes ausgesendete Paket wird die Round-Trip-Time (Umlauf-Zeit) bestimmt. Das ist die Zeit vom Aussenden des Pakets bis zum Empfang der Quittierung. Aus den RTT's der letzten Pakete wird ein gleitender Mittelwert berechnet. Dabei kann es aber zu Problemen mit älteren Paketen kommen. Kommt die Quittierung vom ersten Übertragen nach dem Ablauf des Timer's an, dann könnte der Sender das Paket schon wieder versendet haben. Nun würde die Zeitspanne zwischen 2. Aussenden und der ersten verspäteten Quittierung als RTT interpretiert werden. Solche deutlich zu kleinen Werte würden die Übertragungs-Parameter weiter verschärfen.

### KARN/PARTRIDGE-Algorithmus

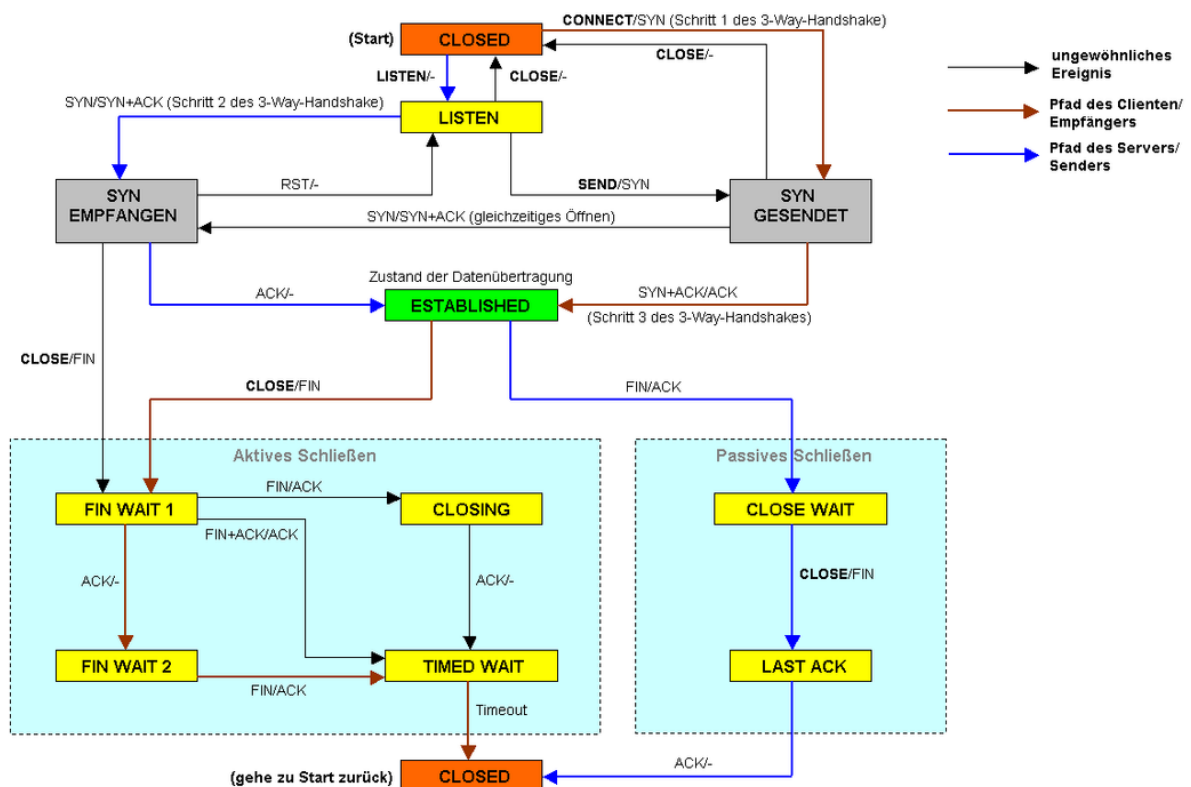
Die Probleme mit dem gleitenden Mittelwert aus der Original-Implementierung des TCP wurden durch KARN und PARTRIDGE mit einem verbesserten Algorithmus (manchmal nur KARN's Algorithmus genannt) behoben. Jetzt werden nur noch die Pakete beachtet, die gesendet und (innerhalb der Zeit) quittiert werden. Zusätzlich erhöht man die Zeit-Schranke bei jeder erfolgreichen Übertragung etwas.

### KAREL/JACOBSON-Algorithmus

Dieser basiert auf dem KARN/PARTRIDGE-Algorithmus. Nun werden die Schwankungen der gemessenen RTT's (Umlaufzeiten) gewichtet in die Berechnung eingeführt. Weiterhin passte der Algorithmus nun auch die Berechnung des Timeout's an.

## TCP-Verwaltung als Endlicher Automat (EA)

siehe dazu auch im Skript (→ [Sprachen und Automaten](#))



Verwaltung der TCP-Verbindung als endlicher Automat  
Q: de.wikipedia.org (Appaloosa)

---

## TCP-Port's

Für den Aufbau und den Betrieb einer TCP-Verbindung werden bei Sender und Empfänger Endpunkte gebraucht. diese nennt man Socket's. Jeder Socket beinhaltet reservierten Speicher-Platz mit mindestens den Eingabe- und Ausgabe-Puffern.

Jeder Socket besteht nur über den Verlauf einer TCP-Verbindung. Ist die Verbindung beendet, wird der Socket freigegeben.

Die Socket's sind durch eine eindeutige Socket-Nummer gekennzeichnet. Sie setzt sich aus der IP-Adresse des Rechners und einer Port-Nummer zusammen. Die Port-Nummer ist eine 16-bit-Zahl – kann also die Werte von 0 bis 65535 einnehmen. Die Port-Nummer wird lokal einer Anwendung zugeordnet.

Ein Port ist somit ein Dienst-Zugriffspunkt (Service Access Point) der (gesamten) Transportschicht.

Damit werden die Port's – als Teil / Erweiterung der Netzwerk-Adresse – Zuordnungs-Punkte zwischen den TCP- und UDP-Verbindungen und den lokalen Client- bzw. Server-Anwendungen.

Zu jeder Verbindung gehören somit immer zwei Port's. Einer beim Sender und einer beim Empfänger. Die Port-Nummern können unterschiedlich sein. Das ist auch meist so, um Client- und Server-Software zu unterscheiden.

Der Ziel-Port ist die Nummer, die der gewünschten Nutz-Anwendung zugeordnet wurde. Für den Verbindungs-Aufbau ist der Quell-Port eine nicht registrierte Port-Nummer. Damit wird jede TCP-Verbindung über ihre Socket-Nummern eindeutig identifizierbar. Ein Server-Socket kann aber mit mehreren (Client-)Socket's verbunden sein. Jeder Client hat dabei eine eigene Verbindung zum Server.

Von 0 bis 1'023 sind die Port-Nummern standardisiert bestimmten Internet- und Netzwerk-Diensten zugeordnet. Man nennt sie auch Well-Known-Port's. Sie dürfen also nicht einfach durch eigene Applikationen verwendet werden. Für die eigenen Programme stehen die Port's mit Nummern über 1'024 zur Verfügung. Aber auch sollte man sich von häufig genutzten Port's fern halten, um keine Inkompatibilitäten zu erzeugen.

Die Nutzung der Port's bis 49'151 muss bei der IANA registriert werden. Die anderen Port's sind völlig frei nutzbar.

Die klassischen TCP/IP-Anwendungen arbeiten mit den Port-Nummern 0 bis 256. Das Betriebssystem UNIX (sowie sein Abkömmling LINUX) nutzen für System-typische Dienste die Port's von 256 bis 1'023.

### **bekannte / häufig genutzte / reservierte / standardisierte Port-Nummern**

Port	Protokoll / Anwendung
23	Telnet
25	SMTP (unverschlüsselt)
53	DNS
80	HTTP

Port	Protokoll / Anwendung

---

Als Schnittstelle zwischen den Applikationen und dem TCP stellt dieses sogenannte TCP-Primitive zur Verfügung. Das sind elementare Operationen / Funktionen, die einen Datenaustausch zwischen Anwendung und TCP ermöglichen. Typische TCP-Primitive sind read, write, request, response, confirm, ...

### ***typische TCP-Primitive***

- request
- response
- confirm
- read
- write
- 

## **Absicherung des Daten-Transport's auf der Transport-Schicht**

### ***grundlegende Sicherheits-Ziele im Internet***

- **Verfügbarkeit**      Rechner soll für andere kontaktierbar sein  
(Denial-of-Service ... Verweigerung des Dienst's)  
Beeinträchtigung z.B. durch DoS-Angriffe
- **Daten-Integrität**    kommen die Daten unverändert beim Empfänger an
- **Vertraulichkeit**    ist die Verbindung vor unliebsamen Mithörern geschützt
- **Authentifikation**    ist der Sender wirklich die bezeichnete Stelle / Person / ...
- **Autorisation**      darf der Client die Daten / Dienste wirklich abfragen  
hat er die notwendigen Rechte

notwendig sind spezielle Absicherungen der Protokolle des TCP/IP-Stapel's  
besonders die Transport-Schicht ist für Absicherungen geeignet  
auf IP-Ebene sehr aufwendig  
Anwendungen können zusätzlich aktiv werden und die Sicherheit teilweise erweitern

### **Transport Layer Security – das TLS-Protokoll**

arbeitet auf der Transport-Schicht

bei Bedarf zusätzlich zwischen Transport-Schicht und Anwendungs-Schicht eingeschobene Schicht

die Nutzlast der TCP-Pakete wird verschlüsselt

das TCP-Paket als solches wird ganz normal übertragen

### **Secure Socket Layer (SSL)**

1994 durch Netscape eingeführt

für den vertraulichen Daten-Transport im www (http)

das dazugehörige Sicherheits-Protokoll war dann https, praktisch http über ssl  
die Funktionalität konnte auch von anderen Anwendungen genutzt werden

---

dadurch große Verbreitung z.B. von online-Banking, e-Shopping, ...  
mit der Version 3 als Standard (TLS) definiert

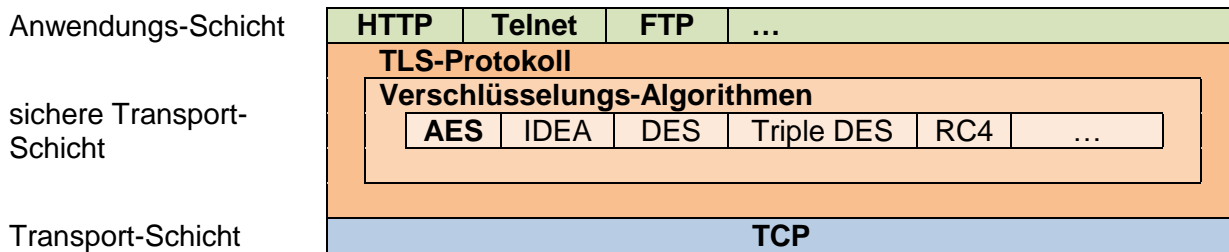
Funktionalität

für viele Dienste verfügbar

bietet den Kommunikations-Partnern eine private (symmetrisch verschlüsselte) Verbindung  
vorher ein per Handshake ausgehandelter Schlüssel-Tausch

Aushandeln erfolgt unter Nutzung von asymmetrischer Verschlüsselung (asymmetrische  
Authentifikation)

es entstehen zuverlässige Verbindungen





---

### **8.3.3.2. UDP – User Datagram Protocol**

bleibt bei Verbindungs-losem Transport von Paketen  
setzt praktisch direkt auf IP auf  
erweitert IP nur um UDP-Port's  
praktisch ist also UDP ein sehr Leistungs-armes Protokoll (oft als "Port-Multiplexing" verpöhnt)

einfach, ungesichert / nicht garantiert  
unzuverlässig (ohne Quittierungs-Mechanismen)  
sehr alt, fast unverändert über die Jahre hinweg  
ermöglicht einfache Zuordnung einer Nachrichten-Übertragung zu einer Anwendung  
übernimmt die Daten von der (über UDP-Port) zugeordneten Anwendung und übergibt sie dem IP  
einfache Frage-Antwort-Kommunikation möglich  
für TCP und UDP festgelegte Port-Nummern können voneinander abweichen, müssen aber gleich sein, wenn sie von beiden Protokollen genutzt werden sollen (z.B. DNS mit Port 53)  
wenn Paket / daten verloren gehen, dann erfolgt einfach eine wiederholte Anfrage

z.B. genutzt für:

- Media-Streaming (Video, VoIP)
- Verluste müssen akzeptabel sein (Anwendung muss Verluste kompensieren)
- wenn keine Segmentierung zu erwarten ist
- Protokolle, bei denen sich ein Verbindungs-Auf- und Abbau nicht lohnt
- 

#### **Vorteile:**

- kein aufwändiger Auf- und Abbau von Verbindungen notwendig
- einfache Frage-Antwort-Kommunikation möglich
- Übertragung von Echtzeit-Daten möglich

#### **Nachteile:**

- keine Sicherheit, dass Daten auch wirklich ankommen
- relativ ungeeignet für die verschlüsselten Daten-Verbindungen

neuere Spezifikationen für UDP sehen auch mehr Absicherungen vor (SRTP, DTLS, ...)

### typische UDP-Anwendungen

- **TFTP**  
Triviale File Transport Protocol      Port 69
- **DNS**  
Domain Name Service      Port 53  
nutzt sowohl TCP als auch UDP
- **NTP**  
Network Time Protocol      Port 123
- **RPC**      Port 111
- **LDAP**      Port 389
- **DHCP**  
Dynamic Host Configuration Protocol      Port
- **RTP**  
Real Time Protocol      Übertragung von Multimedia.Daten
- **SIP**  
Session Initiation Protocol
- **VoIP**  
Voice over IP
- 

### UDP-Datagramm

	4	8	12	16	20	24	28	32
0	(UDP) source port				(UDP) destination port			
32	UDP datagram length				(UDP) checksum			
64	data							

derzeit experimentell: QUIC

soll http über UDP ermöglichen

benutzt eine Zusatz-Schicht zwischen UDP (Transport-Schicht) und der Anwendung(s-Schicht)

Chromium-Browser kann das QUIC schon Client-seitig

derzeit nur wenige Web-Site's, die dieses Protokoll unterstützen (2019 nur rund 3 %)

#### Vorteile:

- weniger Overhead in der Übertragung
- verschränkte Verbindungen möglich (→ http/2)
-

---

**Nachteile:**

- 

HTTP-over-OUIIC wird wohl http/3

## 8.4.1. grundlegende Protokolle

??? ev. noch weiter nach vorne positionieren (hinter IP)

	TCP/IP-Protokoll-Stapel	spezielle Protokolle					
4	Application-Layer						
3	Transport-Layer						
2	Internet-Layer	ARP	RARP	BGP	OSPF	ICMP	IP
1	Link-Layer						

### ICMP – Internet Control Message Protocol (RFC 792 / 1256)

für Fehler-Diagnose

Erreichbarkeits-Test → ping

Aufzeichnen von Zeit-Marken

Erkennen abgelaufener Zeit-Marken

Verwalten der Routing-Tabellen

ermitteln der zulässigen MTU (Maximum Transfer Unit (max. Paket-Länge))

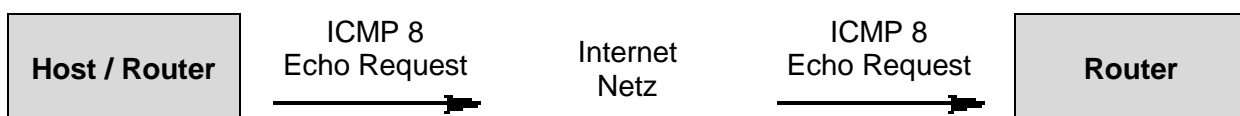
ICMP-Nachrichten werden in IP-Paket gekapselt

im IP-Header wird im Protokoll-Feld eine 1 eingetragen

IP-Nutzlast ist das ICMP-Paket 8 bit Typ-Kennung, 8 bit Code und 16 bit Kontrollsumme  
danach folgt die eigentliche ICMP-Nachricht

Type	Bedeutung
0	Echo Response
3	Ziel nicht erreicht
4	Senderate drosseln
5	Route ändern
8	Echo Request
9	Router Bekanntmachung
10	Suche nach Router
11	Lebenszeit (TTL) überschritten
13	Timestamp-Anforderung
14	Timestamp-Antwort
17	Abfrage Subnetz.Maske

Anforderung durch eine Station:



Antwort des Routers:



auf Kommandozeilen-Ebene mit dem Befehl ping

`ping -a 1 127.0.0.1` (ein Paket an Host-Adresse schicken (hier local host))

unter Windows:

`ping 127.0.0.1`

da die Pakete empfangen und wieder zurückgeschickt word, ist die Adresse erreichbar  
das sollte beim local host auch immer so sein

```
C:\>ping 127.0.0.1

Ping wird ausgeführt für 127.0.0.1 mit 32 Bytes Daten:
Antwort von 127.0.0.1: Bytes=32 Zeit<1ms TTL=128
Antwort von 127.0.0.1: Bytes=32 Zeit<1ms TTL=128
Antwort von 127.0.0.1: Bytes=32 Zeit<1ms TTL=128
Antwort von 127.0.0.1: Bytes=32 Zeit<1ms TTL=128

Ping-Statistik für 127.0.0.1:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (<0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

C:\>
```

nicht existierende Adresse in eigenen Netz  
bzw. Prüfen, ob diese Adresse existiert

Host nicht erreichbar

```
C:\>ping 192.168.100.88

Ping wird ausgeführt für 192.168.100.88 mit 32 Bytes Daten:
Antwort von 192.168.100.141: Zielhost nicht erreichbar.
Antwort von 192.168.100.141: Zielhost nicht erreichbar.
Antwort von 192.168.100.141: Zielhost nicht erreichbar.
Antwort von 192.168.100.141: Zielhost nicht erreichbar.

Ping-Statistik für 192.168.100.88:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (<0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

C:\>
```

existierende Netzwerk-Adresse außerhalb des eigenen Netzwerkes

Host ansprechbar  
Netzwerk-Routing funktioniert

```
C:\>ping 8.8.8.8

Ping wird ausgeführt für 8.8.8.8 mit 32 Bytes Daten:
Antwort von 8.8.8.8: Bytes=32 Zeit=18ms TTL=57
Antwort von 8.8.8.8: Bytes=32 Zeit=17ms TTL=57
Antwort von 8.8.8.8: Bytes=32 Zeit=18ms TTL=57
Antwort von 8.8.8.8: Bytes=32 Zeit=17ms TTL=57

Ping-Statistik für 8.8.8.8:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (<0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 17ms, Maximum = 18ms, Mittelwert = 17ms

C:\>
```

nicht existierende Netzwerk-Adresse außerhalb des eigenen Netzwerkes

```
C:\>ping 192.100.54.34

Ping wird ausgeführt für 192.100.54.34 mit 32 Bytes Daten:
Antwort von 62.155.241.33: Zielnetz nicht erreichbar.
Antwort von 62.155.241.33: Zielnetz nicht erreichbar.
Antwort von 62.155.241.33: Zielnetz nicht erreichbar.
Antwort von 62.155.241.33: Zielnetz nicht erreichbar.

Ping-Statistik für 192.100.54.34:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (<0% Verlust),

C:\>
```

---

## ARP – Adress Resolution Protocol

Umwandeln der IP-adresse aus dem IP-Paket in die zugehörige MAC-Adresse und dann folgt umpacken in ein Paket für das lokale Netz (z.B. Ethernet-Paket)

anzeigen der ARP-Tabelle

`arp -a`

```
C:\>arp -a
Schnittstelle: 192.168.100.141 --- 0xc
Internetadresse    Physische Adresse    Typ
192.168.100.2      c8-0e-14-62-1e-82    dynamisch
192.168.100.142    00-17-c8-84-40-0a    dynamisch
192.168.100.159    fc-aa-14-70-8f-66    dynamisch
192.168.100.170    f0-3f-51-2f-82-43    dynamisch
192.168.100.255    ff-ff-ff-ff-ff-ff    statisch
224.0.0.22         01-00-5e-00-00-16    statisch
224.0.0.252        01-00-5e-00-00-fc    statisch
239.255.255.250    01-00-5e-7f-ff-fa    statisch
255.255.255.255    ff-ff-ff-ff-ff-ff    statisch

Schnittstelle: 192.168.150.1 --- 0xe
Internetadresse    Physische Adresse    Typ
192.168.150.255    ff-ff-ff-ff-ff-ff    statisch
224.0.0.22         01-00-5e-00-00-16    statisch
224.0.0.252        01-00-5e-00-00-fc    statisch
239.255.255.250    01-00-5e-7f-ff-fa    statisch

C:\>
```

## SNMP – Simple Network Management Protocol

UDP-basierter Dienst

für die Konfiguration und Steuerung von Netzwerk-Geräten und deren Kommunikation untereinander

---

## DHCP – Dynamic Host Configuration Protocol

Einbinden neuer Geräte in ein existierendes Netzwerk mit einem DHCP-Server  
dieser ist Teil des Routers oder eines lokalen Servers vergibt ev. eine frei IP-Adresse  
UDP-basierter Dienst

HTTP	IMAP	...	DHCP	DNS	...
TCP			UDP		
IPv4				...	
ARP (Address Resolution Protocol)					
Ethernet		Token-Ring		...	

weitere Einstellungen sind:

Hostname

Default Gateway

zuständiger DNS-Server

unter Windows:

ipconfig

oder

ipconfig /all

unter Linux

ifconfig

## NFS – Network File System

UDP-basierter Dienst

Zugriff auf entfernte Datenspeicher und Arbeitsstationn

---

## TCP-Beobachtung mit Wireshark

Wireshark ist wohl das bekannteste Sniffer-Programm. Es dient der Analyse des Datenverkehrs in Netzwerken. Der Daten-Transfer kann aufgezeichnet werden und graphisch gestützt angezeigt werden.

<https://www.wireshark.org>

schneidet den gesamten Netzwerk-Verkehr mit  
zur Erinnerung: im Netz werden vielfach (z.B. beim WLAN) alle Daten-Pakete an alle versendet  
jede Netzwerk-Endstelle nimmt dann aber nur die Pakete zur Weiter-verarbeitung auf, die an diese Endstelle adressiert sind



Die Analyse und die Veränderung von fremden Daten ist strafbar.  
Das Tool Wireshark darf also nur innerhalb des eigenen Netzes genutzt werden und damit nur eigene Daten analysiert werden.

in aufgezeichneten Protokollen kann dann gefiltert werden  
Eingabezeile unter der Symbol-Leiste  
grün bedeutet, dass es eine gültige Anfrage (ein gültiger regulärer Ausdruck (s.a. → ) ist  
bei rotem Hintergrund fehlen noch Info's oder es liegt ein Syntax-Fehler vor

hierrarchische Struktur  
z.B.:

tcp.flags	zeigt alle Pakete mit irgendwelchen gesetzten Flag's an
tcp.flags.syn==1	sucht Pakete mit gesetztem SYN-Flag heraus
ip.src==192.168.01	sucht nach allen Einträgen mit der angegebenen IP-Quell-Adresse
ip.dst==...	entsprechend für Ziel-Adresse
ip.addr==...	für Ziel- und Quell-Adresse

zwei Anfragen (reguläre Ausdrücke) lassen sich auch mit **and** bzw. **&&** verbinden  
entsprechend gilt für die Alternative **or** bzw. **||**  
auch Klammern ( ) sind möglich



---

### Aufgaben:

- 1. Zeichnen Sie für eine Schnittstelle (auf der Datenverkehr passiert) 1 bis 2 min den Netzwerkverkehr mit Wireshark auf! Erfüllen Sie währenddessen die Aufgaben 2 bis 5!*
- 2. Ermitteln Sie z.B. mit dem Konsolen-Programm ipconfig die Konfiguration Ihrer lokalen Netzwerk-Anschlüsse!*
- 3. Pingen Sie ebenfalls in der Konsole einmal den Google-Server 8.8.8.8 an!*
- 4. Rufen Sie einmal die Google-Webseite auf und suchen Sie nach Ihrer Schul-Website!*
- 5. Downloaden Sie eine Datei über den folgenden Link ! (Verwenden Sie dazu die Nutzer-Kennung: Name: Mustermann (geheimes) Kennwort: Mu5B3rm4nn)*
- 6. Filtern Sie die TCP-Pakete heraus, die an Ihre lokale Netzwerk-Adresse gegangen sind!*
- 7. Welche ARP-Pakete wurden im Netz transportiert? Erläutern Sie deren Funktion!*
- 8. Schauen Sie sich den (S)FTP-Verkehr mal genauer an! Welche Informationen können Sie dem Protokoll entnehmen? Bedenken Sie, dass Sie sich auch gerade in einem öffentlichen WLAN befinden könnten!*

## 8.4. Internet-Anwendungen

von 1969 bis 1993 nur sehr schwaches, lineares Wachstum der Internet-Nutzung  
 in dieser Phase kam auch der erste PC (von IBM) 1981 dazu; vorher nur HomeComputer  
 (Spiele. ...; Kommunikation über BTX oder Modem)  
 echter Start des Internet's 1989 mit dem www / http  
 1990 Abschaltung des ARPANET (durch TCP/IP abgelöst) und nun auch starke kommerzielle Nutzung des Internet's  
 ab 1993 Nutzung auch durch Bevölkerung (erster (graphischer) Browser)  
 ab 1993 bis 2007 potientiell bis expotentiell Wachstum  
 Suchmaschinen waren yahoo, fireball, altavista, ...  
 ab 1998 google als Suchmaschine mit deutlich effektiveren und Anwendungs-freundlicheren Algorithmen  
 2001 Start von wikipedia  
 dann ab 2003 soziale Medien mit facebook und auch Start des "web 2.0" als Mitmach-Internet  
 ab 2007 wieder eher lineares Wachstum, allerdings mit sehr rasanten Anstieg  
 (pro Jahr jetzt rund 500'000'000 Nutzer / Nutzungen mehr)  
 ab 2014 starke Verbreitung von Musik- und Video-Streaming  
 zusätzlich 2015 IoT (Internet of Things)

setzen alle auf der Internet-Schicht auf

Kommunikation zwischen z.B. TCP und der Anwendung (z.B. das eMail-Programm Thunderbird oder Outlook)

Anwendungs-Schicht stellt universelle Schnittstellen für die Anwender-Programme bereit

Internet-Anwendungen in diesem Sinne sind also nicht die Nutz-Programme, wie z.B. Browser (z.B. Internet Explorer, Firefox, Chrome, Opera, ...) oder eMail-Programme (Outlook, Thunderbird, ...). Hier verstehen wir Internet-Anwendungen als grundsätzliche Kommunikations-Arten, wie z.B. eMailing oder das world wide web (www / http). .

Schicht 4	Anwendung	HTTP	IMAP	...	DHCP	DNS	...
Schicht 3	Transport	TCP		UDP			
Schicht 2	Internet	IPv4			...		
Schicht 1	Netzwerk-Zugriff	ARP (Address Resolution Protocol)					
		Ethernet		Token-Ring ...			

## DNS – Domain Name Service

Umwandlung von www-Adressen in IP-Adressen

TCP- und UDP-basierter Dienst

eher unbekannter Dienst, läuft im Hintergrund, ist aber für die Mensch-Maschine-Kommunikation sehr wichtig

HTTP	IMAP	...	DHCP	DNS	...
TCP		UDP			
IPv4			...		
ARP (Address Resolution Protocol)					
Ethernet		Token-Ring ...			

Die wenigsten Menschen möchten sich für die Suchmaschine google die eigentliche Adresse 172.217.22.67 merken. Pingt man google an, dann erhält man die exakte IP-Adresse.

```
C:\Users\drews>ping google.de

Ping wird ausgeführt für google.de [172.217.22.67] mit 32 Bytes Daten:
Antwort von 172.217.22.67: Bytes=32 Zeit=17ms TTL=57
Antwort von 172.217.22.67: Bytes=32 Zeit=16ms TTL=57
Antwort von 172.217.22.67: Bytes=32 Zeit=16ms TTL=57
Antwort von 172.217.22.67: Bytes=32 Zeit=16ms TTL=57

Ping-Statistik für 172.217.22.67:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
    Minimum = 16ms, Maximum = 17ms, Mittelwert = 16ms
```

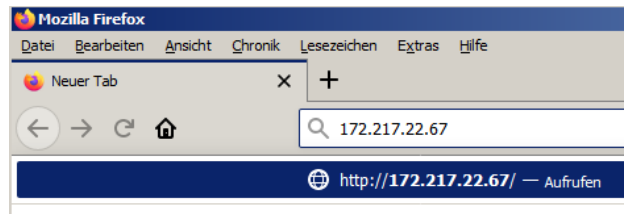
Genau lässt sich jede IP-Adresse zu einer Website ermitteln.

Die Adressen sind aber nicht besonders handlich und irgendwann wird ja auch das gesamte Netz auf IPv6 umgestellt sein, dann wird es wohl unmöglich solche Adressen fehlerfrei in den Browser einzugeben.

Der DNS übernimmt die stupide Arbeit die schönen handlichen Text-Adressen (Domain-Namen), wie eben www.google.de oder de.wikipedia.org in die generischen IP-Adressen zu übersetzen.

Auch die umgedrehte Arbeit wird von DNS erledigt. So kann man eben auch herausbekommen, wer hinter 172.217.22.67 steckt.

Hier ist ein Browser die beste Wahl zum Erkunden. Allerdings funktioniert die Anzeige nur, wenn die Adresse über einen Web-Server verfügt.



**Aufgaben:**

***1. Ermitteln Sie die IP-Adressen für die folgenden Website's!***

- |                     |                     |                     |
|---------------------|---------------------|---------------------|
| a) zdf.de           | b) www.bsi.bund.de  | c) www.atomzeit.eu  |
| d) de.wikipedia.org | e) en.wikipedia.org | f) dk.wikipedia.org |
| g) open.hpi.de      | h)                  | i)                  |

***2. Wer steckt hinter den folgenden IP-Adressen? Gibt es sie überhaupt?***

- |                   |                   |            |
|-------------------|-------------------|------------|
| a) 52.178.155.90  | b) 212.227.247.48 | c) 8.8.8.8 |
| d) 91.198.174.192 | e) 192.168.0.1    | f) 2.2.2.2 |
| g) 127.0.0.1      | h) 153.384.245.16 | i) 1.1.1.1 |

***3. Vergleichen Sie die (Telefon-)Auskunft mit dem DNS!***

**für die gehobene Anspruchsebene:**

***4. Prüfen Sie, ob Ihr Netzwerk auch mit IPv6 umgehen kann! Wenn JA, dann ermitteln Sie die IPv6-Adresse von google.de!***

Dienst wurde 1983 auf Vorschlag von MOCKAPETRIS eingeführt. Domain-Namensraum ist hierarchisches System mit einer Baum-Struktur



Zentrale vergabe durch ICANN bzw. von ihr delegiert an lokale Domain-Registaturen. Für Deutschland ist das die DE-NIC. Über die denic.de kann man auch Inhaber abfragen.

Im Internet sind bestimmte Domainname-Server (DNS-Server) aktiv. Früher mussten die Adressen der Name-Server aktiv beim Router oder dem vernetzten PC eingegeben werden. Heute erfolgt das meist automatisch über DHCP.

Beim Domain-Name-Server wird eine Tabelle geführt, die IP-Adresse und Domain-Name gegenüberstellt. Damit die Tabellen nicht unendlich groß werden, was ja mit längeren Bearbeitungs-Zeiten verbunden wäre, verwaltet jeder DNS-Server eine Zone. I.A. sind diese Zonen nicht überlappend. D.h. jeder Server kennt einen Teilzweig des Domain-Name-Baum's. Der Client (bzw. das Anwendungs-Programm) stellt beim DNS-Server die Anfrage mit dem aufzulösenden Domain-Namen z.B. www.google.de. Der DNS-Server schaut in seine Tabelle und liefert bei einem vorhandenen Eintrag die passende IP-Adresse zurück an den Client.

Kennt der DNS-Server den angefragten Eintrag nicht, dann gibt er die Anfrage an einen übergeordneten DNS-Server weiter. Dieser prüft nun, ob der angefragte Domain-Name in seiner Tabelle ist. Wenn nicht, dann wird nachgesehen, ob eine übergeordnete Domäne zuordnenbar ist. In dem Fall existiert ein anderer DNS-Server in seiner Tabelle, an der er wieder die Anfrage weiterreichen kann. Der Domain-Name wird dann von einem anderen DNS-Server verwaltet. Hat dieser einen passenden Eintrag, liefert er diesen zurück. Ist auch hier der Name wird der Hierarchie weiter gefolgt.

Über das DNS-System sind auch alternative – versteckte – Domain-Namen möglich. Diese können sogar die gleichen Namen, wie die Domain's aus dem "normalen" Internet, benutzen. Was es braucht ist nur ein eigener DNS-Server. Diese kann die – im "normalen" Internet – nicht anwählbaren Adressen verwalten. Sachlich lässt aber jede "verborgene" IP-Adresse direkt anwählen, nur eben nicht über den zugeordneten Domain-Namen. Man spricht auch vom Dark Net, Deep Net oder dem Hidden Net.

Unsichtbar sind die Geräte aber zuerst einmal nur für das "normale" DNS-System. Über ihre IP-Adresse sind sie jederzeit für jeden Internet-Nutzer erreichbar.

Mit Hilfe eines Caching-System's wird der Daten-Verkehr über andere DNS-Server reduziert. Hat ein DNS-Server von einer früheren Anfrage ein im unbekanntes Paar IP-Adresse und Domain-Name von einem übergeordneten DNS-Server zurückbekommen, dann merkt er sich diesen in seinem DNS-Cache. Dieser Speicher ist eher flüchtig angelegt. Ältere Anfragen werden weiter hinten angeordnet und irgendwann gelöscht. Bei neueren Anfragen besteht ja die berechnete Chance, dass bald wieder eine gleichlautende Anfrage kommt (→ Lokalitäts-Prinzip). Durch den DNS-Server ist dann nur ein kurzer Kockup notwendig und er hat den passenden Tabellen-eintrag gefunden. Der neuere Eintrag wird quasi weiter vorn in der Tabelle gehalten. Für alle Cache-Einträge gibt es eine TTL (Time of Live). Spätestens nach Ablauf dieser Zeit, wird der Cache-Eintrag gelöscht. Bei erneuten Benutzen eines Cache-Eintrages wird dessen Lebenszeit neu gestartet.

Kommt es mal zu einem Ausfall eines DNS-Server's, dann lädt dieser beim Neustart nur seine Basis-Tabelle und baut seinen Cache ganz von vorne wieder auf.

.edu	
.com	
.gov	
.mil	
.org	

.at	Östereich

.info	
.biz	
.tv	
.house	

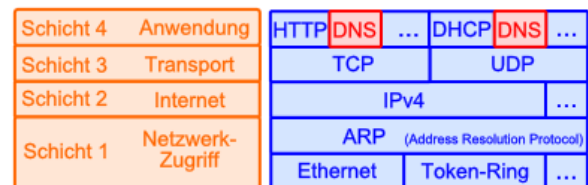
## DNS-Tabelle im Detail

Domain Name	TTL	Klasse	Typ	Wert
-------------	-----	--------	-----	------

Beispiele:

60  
86400

Typ		
<b>SOA</b>	Start of Authority	Parameter für die zugehörige Domain
<b>A</b>	Host IP	IP-Adresse
<b>MX</b>	Mail Exchange	Domäne für eMail
<b>NS</b>	Name Server	Nameserver für Domäne
<b>CNAME</b>	Kanonischer Name	Domänenname
<b>PTR</b>	Pointer	Eintrag für reverses DNS Lookup
<b>HINFO</b>	Host Info	CPU / OS
<b>TXT</b>	Text	nicht interpretierter Text



## electronic Mail – eMail – POP / SMTP / IMAP

1971 erste eMail verschickt; in Deutschland die erste eMail 1983  
 entwickelt von Ray TOMLINSON (1941 – 2016)  
 ist praktisch die erste "Killer"-Anwendung gewesen, hat dem Internet zu einem ersten großen Durchbruch verholfen  
 auch heute noch das am weitesten verbreitete Kommunikations-Mittel (der Spam hilft hier natürlich auch noch nach)  
 2019 rund 4 Mrd. eMail-Nutzer (bei rund 1,75 eMail-Account's pro Nutzer)  
 mit rund 280 Mrd. eMails am Tag

SMTP – Simple Mail Transfer Protocol  
 TCP-basierter Dienst

Text-basierte Informations-Übertragung

Schicht 4	Anwendung	POP	SMTP	IMAP	...	DNS	...
Schicht 3	Transport	TCP				UDP	
Schicht 2	Internet	IPv4				...	
Schicht 1	Netzwerk-Zugriff	ARP (Address Resolution Protocol)					
		Ethernet		Token-Ring		...	

Aufbau einer eMail-Adresse

@-Zeichen als Trenner von Postfach und Domain-Name

**Benutzername** @ **Domain**  
**eMail-Fach** **Server**  
**IP-Adresse**

Beispiele:

info @ lern-soft-projekt.de  
 Hr.Kl.Mustermann @ web.de

Das ursprüngliche eMail-System war an spezielle Applikationen, wie z.B. Outlook, Thunderbird, Mail usw. gebunden. Sie mussten / müssen installiert werden. Die Daten (geschriebene und erhaltene eMails) sind nur auf diesem Rechner verfügbar.  
 Diese Programme setzten direkt auf die Anwendungs-Schicht des TCP/IP-Stck's auf.

### **Komponenten eines eMail-System's**

- **Internet** Medium
- **User-Agent** Nutzer-Beauftragter  
**UA** je Nutzer 1 UA
- **Message-Transfer-Agent** Nachrichten-Übertragungs-Beauftragter  
**MTA**

MTA's sind im Internet verteilt auf verschiedenen Servern und ermöglichen den Transport von eMail untereinander  
 gesamtes System wird Message Transfer System genannt

## SMTP – Simple Mail Transfer Protocol

mit diesem Protokoll arbeiten die UA's  
regelt die Kommunikation zwischen zwei / den MTA's  
normalerweise werden nur alphanummerische Zeichen übertragen (eMail ist klassischer Text-basierter Dienst)

HTML ist möglich, da auch Text-basiert (lange Zeit als problematisch eingestuft, weil so das Einschleusen von Schad-Software sehr einfach möglich war)  
für Multimedia-Daten muss eine Umsetzung gemacht werden (→ MIME)

Der SMTP-Handshake baut zuerst eine Verbindung zwischen dem Client-MTA und der Server-MTA auf. Dazu wird zuerst über Telnet die Existenz eines Server's geprüft. Nach der Bestätigung wird mit HELO die eigentliche SMTP-Kommunikation begonnen. Der Client-MTA gibt dabei seinen eigenen Namen weiter.

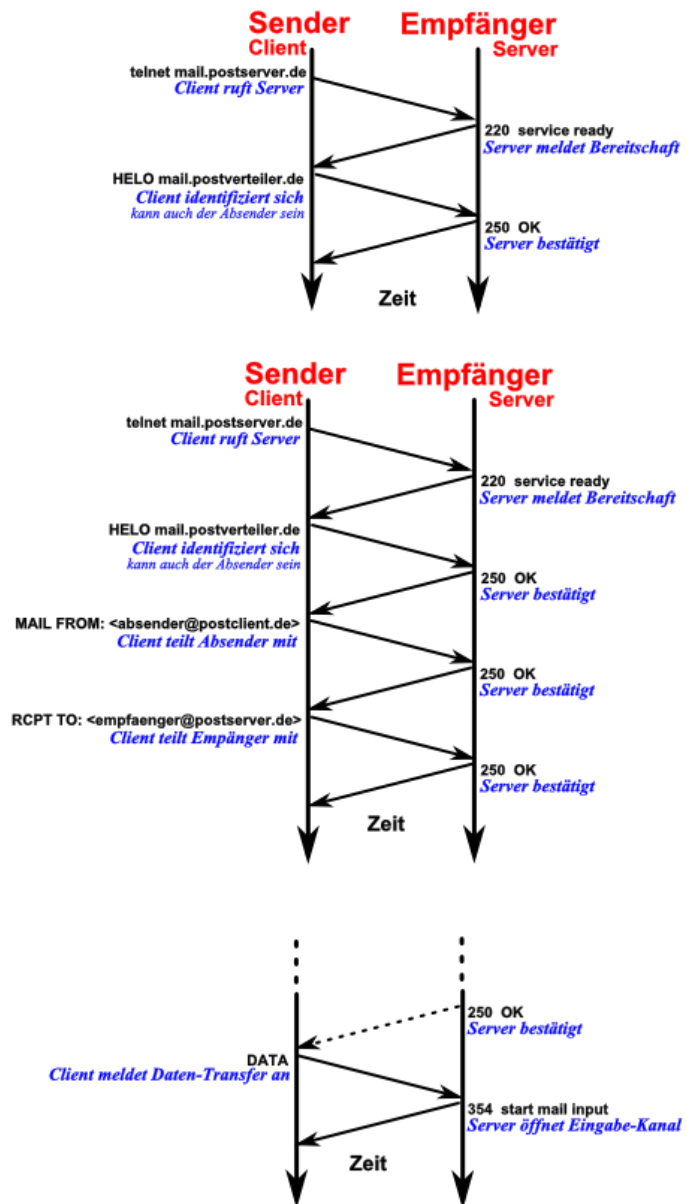
Der Client-MTA kann der originale Absender sein, oder eben auch eine beliebige Zwischen-Station.  
Nun tauschen die MTA's die Absender- und Empfänger-Adressen aus.

Dafür gibt es jeweils spezifische Nachrichten-Köpfe. Hier müssen die eMail-Adressen (zum mindestens, die des Empfänger's) auch stimmen, da sonst keine Zustellung möglich ist.

Der Server-MTA bestätigt jedesmal mit einem OK-Kommando.

Mit dem Empfang der Bestätigung durch den Client kann dieser nun den Daten-Transfer ankündigen.

Der Server-MTA ermöglicht jetzt den Empfang einer längeren text-Nachricht.



Die vom Client übertragene eMail ist nun ein eigenes Text-Objekt. In ihm sind neben dem eigentlichen Text-Körper auch noch als Header (Kopf-Teil) Absender und Empfänger angegeben.

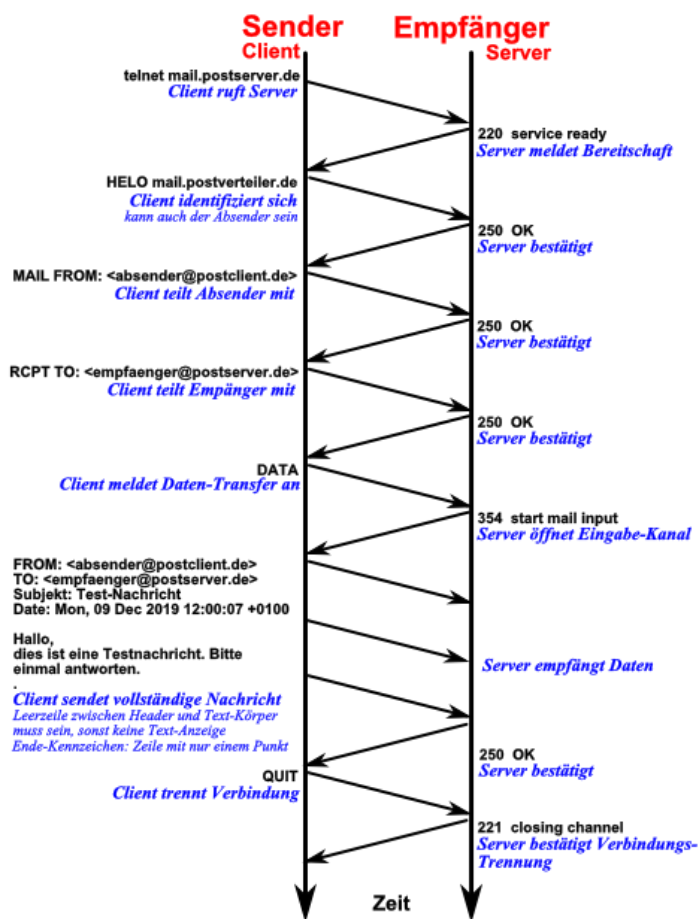
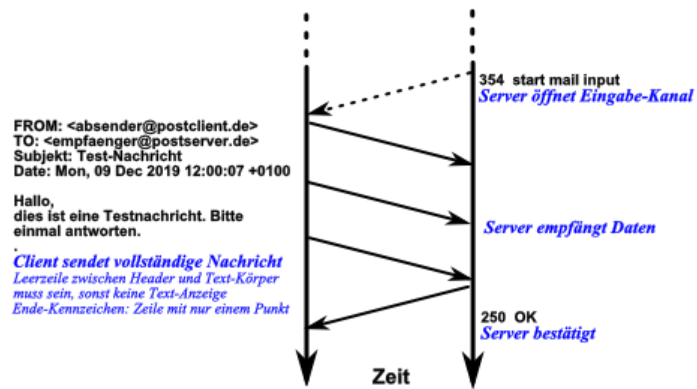
Interessanterweise sind diese Kopf-Daten weitgehend manipulierbar. Erst dadurch ist das nervige Spammen erst möglich. Der Spammer kann hier nämlich alles eintragen, und damit auch seine wahre Identität verschleiern.

Das Ende der Text-Nachricht ist durch eine Zeile gekennzeichnet, die nur einen Punkt enthält.

Der Server quittiert den Empfang der eMail.

Zum Abschluss initiiert der Client die Trennung der Verbindung, was der Server mit einem speziellen Signal bestätigt.

Letztendlich handelt es sich wieder um ein schönes Beispiel für ein Handshake-Verfahren.



## Internet Media Type: MIME – Multiple Internet Mail Extension

auch Multipurpose Internet Mail Extension erweiterter Daten-Format von eMails, um unterschiedlichste Daten-Typen im eMail-System übertragen zu können. Übersetzt beliebige Daten-Dateien (! Vorsicht es gehen auch EXE) in alphanumerische Zeichen. Diese werden dann beim Empfänger zurückgewandelt.

Bei Endstellen müssen die Chiffrierung kennen, sonst MIME-Typ-Fehler. Die Einbearbeitung erfolgt dann über ein geeignetes Multimedia-Programm oder PlugIn.

Je LAN bzw. Email-Einheit ist ein eMail-Gateway notwendig. Ist praktisch der eMail-Server mit den vielen eMail-Fächern.



---

Das Gateway ist praktisch das "lokale" Postamt. Die Gateway kommunizieren als MTA's über das SMTP. Die eMail-Clients (also die eMail-Programme) auf den Endgeräten fragen die eMail's mittels POP (Post Office Protocol) ab. Aktuelle Version ist POP3.

Durch POP wird:

die Anmeldung beim eMail-Gateway (POP3-Server)

herunterladen der eMail und danach sofortiges Lösen auf dem Server

Beim Senden ist das lokale eMail-Programm aber kurzzeitig auch ein MTA und muss deshalb auch SMTP können und einen nächsten SMTP-Server kennen.

Als Verbesserung zum POP kam dann das IMAP (Interactive Mail Access Protocol). Bei POP werden immer alle Teile einer eMail heruntergeladen und lokal gespeichert. Das ist besonders bei unerwünschten und gefährlichen Anhängen nicht sinnvoll. Weiterhin ist kein paralleles Arbeiten mit mehreren Konten möglich. Ein Postfach muss nach dem anderen abgefragt werden. Desweiteren ist ja nach einem Kontakt beim POP3-Server durch das Herunterladen der eMail auf dem einen Gerät keine Möglichkeit mehr da, automatisch eMails gleichwertig auf mehreren Geräten zu bearbeiten.

Beim IMAP werden zuerst nur die Kopf-Daten einer eMail (al'a POP) heruntergeladen. Erste wenn man sich für ein Lesen entscheidet, dann wird der Rest mit all den Anhängen heruntergeladen.

Nutzer können jetzt beliebig viele eMail-Programme an verschiedensten Orten (Firma, Zuhause, Unterwegs) benutzen. Alle können über den gleichen Daten-Bestand verfügen.

Auf den IMAP-Server wird jetzt auch die Möglichkeit geboten, die eMails in verschiedenen Ordnern zu speichern / einzusortieren. Die Speicherdauer ist jetzt nicht mehr begrenzt. Als limitierenden Faktor tritt nur die gebuchte / zugeordnete Speicher-Größe in den Vordergrund.

Diese Bindung an ein oder mehrere Gerät hat sich als Nachteil (in unserer mobilen Kommunikation) herausgestellt. Nicht jeder kann seine eMail's immer nur auf einem Gerät bearbeiten. Denken wir nur an uns, wenn wir (ohne Handy) auf Reisen sind. Der Wunsch eMail's Orts-unabhängig zu verwalten, hat schnell zugenommen. Deshalb bieten die meisten Mail-Anbieter auch Web-basierte Zugänge zu den Postfächern an. Die eMail-Anbieter web.de, gmx, outlook.de und yahoo sind hier sicher die bekanntesten Beispiele. Bei ihnen wird die Nutzer-Kommunikation über den Browser – also das http-Protokoll – erledigt. Die eMail-Server untereinander benutzen aber immer noch das klassische eMail-Protokoll.

eMail-Programme leisten heute:

- Erzeugen einer Nachricht (Composition)
  - Text-Editor
  - Zusammenstellung des Header's
  - Verknüpfen von Anhängen an die Mail
- Zustellen einer Nachricht (Transfer)
  - Anmeldung an der Servern
  - Verbindungs-Auf- und -Abbau
  - Nachrichten-Übertragung
- Benachrichtung über die Zustellung / Mail-Status
  - Rückmeldungen zum Versand / Empfang
  - Erkennen von Fehlern, ...
- Anzeigen der Nachrichten (Displaying)
  - Anzeigen der Texte
  - Übersetzung von HTML-Tag's
  - Umsetzen der MIME-Daten
- Speichern / Archivieren von Nachrichten

- 
- sicheres Speichern der Nachrichten
  - Löschen von Nachrichten
  - ...

Beispiele:

- microsoft Outlook
- mozilla Firefox
- Apple Mail
- microsoft Mail
- 

relativ hohes Gefahren-Potential:

- Absender-Daten können manipuliert werden (eMail-Spoofing (→ CEO-Attacke)
- Phishing
- bösartige Anhänge (EXE oder getarnte EXE, die sich als andere Datei ausgeben (z.B. PDF)

## **File Transport Protocol - FTP**

## **Instant Messaging / Chat -**

vieles als Erweiterung der elektronischen Post

## **Hypertext Transfer Protocol - http / www**

www – world wide web  
ein Bereich des Internet's  
praktisch Nutzung des http (Hypertext Transfer Protocol)

das was viele populär als Internet verstehen  
eigentlich nur ein Dienst unter vielen  
neben eMail wohl der am häufigsten genutzte Internet-Dienst  
TCP-basierter Dienst

---

Web-Seiten bieten heute (praktisch fast) alle Informationen:

- Informationen zu Firmen / Institutionen
- Suche von Seiten
- eShopping
- online-Lexika
- online-Banking
- Paket-Verfolgung
- soziale Medien / Communities
- Unterhaltung / Multimedia / Streaming
- Vertretungs-Pläne
- Blogs / Nachrichten
- Reise-Portale
- Preis- und ... -Vergleiche
- eLearning
- eMailing über Browser
- InstantMessaging
- Chat
- Routen-Planung / Navigation
- ...

notwendig ist Client-Programm → Web-Browser od. kurz nur Browser genannt  
Beispiele: Internet Explorer, Edge, Safari, Firefox, Opera, Chrome, ...  
dient zur Anzeige und zum Navigieren auf Internet-Seiten (http-Seiten)

bereitgestellte Daten (Seiten-Inhalte) müssen in einem definierten Format angeboten werden  
Text-basierte, getagte Daten → Hypertext- bzw. Hypermedia-Format  
ursprüngliches Ziel war die flexible Darstellung von (zuerst einmal nur) Texten auf verschiedensten Computersystemen und Bildschirm-Arten und -Größen. Beim Start des www waren auch noch monochrome und / oder kleine Monitore üblich. Große farbige Monitore kosteten viel Geld.

Browser baut zur eingegeben / angeklickten Adresse (Link / URL) eine TCP-Verbindung auf  
Seite-Inhalt wird abgerufen (als getagter Text) wird angezeigt  
ev. enthalten die Seiten wieder Zeiger auf andere Seiten (Adressen anderer Seiten oder auch Zeiger / Pointer)  
dargestellt als Link (üblicherweise als unterstrichener Text)  
heute volle Intergration von Multimedia und Programmiersprachen (Skript-Sprachen)

URL-Uniform Resource Locator  
ist die Internet-Adresse einer Webseite

`http://www.subdomain.domain.topleveldomain/pfad/datei`

http-Dienst arbeitet Verbindungs-bezogen nach dem Client-Server-Prinzip mit mehrfachen Handshake

### **Markierungssprache HTML – Hypertext Markup Language**

HTML-Datei enthält sowohl die Daten ((Text-)Informationen) als auch Steuer-Wörter für den Browser. Steuer-Wörter werden Tag's genannt. Sie sind in spitze Klammern (< >) eingeschlossen. Tag's sind meist gepaart, dabei gibt es einen einleitenden Tag und einen beendenden Tag. Beide besitzen den gleichen Text innerhalb der spitzen Klammern. Der beendende Tag hat einen zusätzlichen Schrägstrich (Slash, /) vor dem Schlüsselwort.

---

z.B. **<B>** .... **</B>** für Fett-Anzeige des eingeschlossenen Textes

einzelne Tag's haben keinen beendenden Tag, z.B. **<BR>** für Break als Zeilen-Umbruch.

diverse Tag's besitzen im einleitenden Tag die Möglichkeit zusätzliche Optionen festzulegen im abschließenden Tag sind dann nicht mehr zulässig und auch nicht zugelassen.

Gute Browser sind sehr Fehler-tolerant. Werden bestimmte – vor allem abschließende – Tag's nicht exakt gesetzt, dann krigieren sie diese Fehler.

durch Tag's lassen sich Multimedia-Inhalte einbinden, Diese werden binär nachgeladen

## Exkurs: Aufbau einer HTML-Datei

HTML-Dateien sind sogenannte "Hyper Text Markup Language"-Dateien. Das bedeutet übersetzt Hypertext-Markierungs-Sprache.

In einer HTML-Datei ist der originale Text in lesbarer Form enthalten. Die besondere Hervorhebung z.B. von fett geschrieben Text-Teilen oder Überschriften wird durch sogenannte Tag's erreicht. Tag's werden in spitze Klammern notiert. Die Browser werten diese Tags aus und stellen die Texte dann entsprechend dar. Die meisten Tag's bestehen aus einem einleitenden Tag und einem beendenden. Beide besitzen den gleichen Innentext, nur dass der End-Tag noch einen einleitenden Schrägstrich dazu enthält

Steht in einem HTML-Text die folgende Sequenz:

```
...
Der Text wird ab hier <b> fett gedruckt: Hallo, hallo </b>.
...
```

dann bewirkt die eine fettgedruckte Ausgabe des in die Bold-Tag's eingeschlossenen Wörter. Es ergibt sich also die Ausgabe:

Der Text wird ab hier **fett gedruckt: Hallo, hallo.**

Im Folgenden zeigen wir eine sehr, sehr einfache HTML-Datei. Diese dient nur dazu, um das Prinzip zu verdeutlichen und die wesentlichen Bereiche zu besprechen. Rechts daneben sind einige kurze Erläuterungen. Die Datei kann aber mit einem Text-Editor erstellt werden. In einem beliebigen Browser wird sie dann – wie eine Webseite – angezeigt.

```
<html>
<head>
<title> Startseite </title>
<meta content="Homepage" >
</head>

<body>
<h2> meine Homepage </h2>

Hallo, willkommen!<br>
Dies ist die Seite von
<b> dein Name </b>

</body>
</html>
```

Start des HTML-Textes  
Beginn des nicht-sichtbaren Kopf-Bereiches  
Titel für das Browser-Fenster  
Themen-Aufzählung  
Ende des Kopf-Bereiches

Beginn des anzeigbaren Inhaltes  
eine Überschrift (Gliederungsebene 2)

normaler Text mit einem Zeilenumbruch

fett gedruckter Text

Ende des Inhalts-Bereichs  
Ende des HTML-Textes



Die Ausgabe der HTML-Datei ist sehr einfach gehalten. Für eine optisch aufwendigere Internetseite sind neben HTML auch noch andere Technologien notwendig.

Wer sich über – zumindestens über Teile – einer beliebigen Internetseite informieren möchte, kann das praktisch in jedem Browser tun. I.A. gibt es beim Klick auf die Webseite und beim Aufruf des Kontext-Menüs einen Punkt, der den Quelltext anzeigen lässt.

Für moderne Internetseiten benutzt man spezialisierte Programme, die Text- und Layout-Vorgaben automatisch in HTML usw. umsetzen.

---

## Suchmaschinen

### soziale Netzwerke

auf der Basis des http-Systems

Möglichkeit:

der einfachen Selbst-Darstellung (→ Profil)

schnelle Information, einfache(r) Austausch / Kommunikation → praktisch immer aktuell

Ort- und Geräte-unabhängig zu agieren

Kommentare hinzufügen

Nutzer-Gruppen zu bilden

Multimedia-fähig

leichte Zugänglichkeit

Vorteile:

- starke Vernetzung von Personen, Organisationen und Geräten
- hohes Maß an möglicher Interaktivität
- Zusammenarbeit (Kollaboration) möglich
- Nutzer sind Produzent und Konsument
- ...

Nachteile:

- Entkopplung von Person und Profil – größere Anonymität → verleitet zum Vergessen der guten Erziehung
- viele persönliche und Personen-bezogene Daten im Umlauf
- schnelles Umschwenken von den (Haupt-)Inhalten zu Formen / Neben-Inhalten
- mögliche Zensur (Nutzer sollen positiv / wohlwollend gestimmt werden)
- Macht-Mißbrauch durch Besitzer / ...
- ...

jeder kann jetzt auf natürliche Weise teilnehmen (Mitmach-Netz) ohne größere Computer-Kenntnisse

leicht bedienbar

---

## **Formen**

- **Blog** Diskussions-System zu geposteten / veröffentlichten Texte, (Bilder,) und / oder Podcast's
- **Forum** Frage-Antwort-Diskussions-Plattformen
- **Wiki** z.B.: wikipedia  
interaktive, Hypertext-basierte Lexika
- **Media Sharing** z.B.: YouTube, Instagramm, ...
- **soziale Netzwerke** z.B.: Facebook, Stayfriends, ...
- **Lern-Plattformen  
MOOC's** Massive Open Online ...

## **Komponenten eines / vieler sozialen Netzwerke:**

- Profil
- Fotoalben
- Messaging
- Kommentar- / Feedback-Tools
- Apps
- Timeline

## **Wiki's**

ganz besonders wikipedia haben zu einem neuen Lexika-Verständnis und zu geänderter Benutzung geführt (heute copy-and-paste und vielfach keine semantische Reflexion des Inhalt's)

inhaltliche Tiefe sehr heterogen, da keine oder schwache redaktionelle Betreuung; hierarchisches System an Betreuern; gewählte Administratoren achten auf die Einhaltung von Regeln  
Seiten-Inhalte können manipuliert werden

2001 gegründet

heute betrieben von der Wikimedia Foundation (Non-Profit-Organisation)

derzeit rund 50 Mio. Artikel (rund 2,5 Mio. deutsch-sprachig)

pro Artikel werden durchschnittlich rund 50 Bearbeitungen eingebracht

## **8.4.x. erweiterte Protokolle**

??? Einordnung

## **Media-Streaming**

mit 2019 Gleichzug von Media-Streaming mit der TV-Nutzung  
deutlich stärkerer Anstieg des Streaming's als Abfall der TV-Nutzung

---

Streaming verändert Art der Daten-Bereitstellung und –Nutzung  
traditionell wurden Daten immer erst (vollständig) heruntergeladen und dann genutzt / angezeigt  
Verfahren z.B. für große Daten-Menge und Life-Übertragungen nicht geeignet

beim Streaming wird nach einem kurzen Vorlauf des Download's (Datenstrom's) gleich auch mit der Anzeige begonnen

bei normalen Internet-Verbindungen immer kleiner Vorlauf des Downloadens

Daten werden in einem Puffer gespeichert und nach der Nutzung wieder von neuen Daten überschrieben

die Daten aus dem Internet kommen unregelmäßig an, der Puffer funktioniert aber als Warteschlange (FIFO-Speicher), wobei die Entnahme i.A. geringer ist als die Einspeicherung (Entnahme ist langsamstes Element der Kette)

durch die Nutzung unterschiedlicher Aufösungen lassen sich die Daten-Mengen beeinflussen  
aus dem Puffer werden die Daten immer gleichmäßig vom Media-Player entnommen und angezeigt

auf Servern gespeicherte Daten können Wusch-gerecht übertragen werden.  
Sprünge und Vor-/Zurück-Spulen ist möglich

Live-Streaming

Daten werden praktisch Zeit-gleich mit der Nutzung erzeugt

## **Mediatheken, aNetflix und Co**

Beispiele:

- Mediatheken von:
  - ARD, ZDF und CNN
  - TVNOW (RTL, VOX, n-tv, ...)
  - Joyn (Pro7, Sat1, sixx, DMAX, ...)
- soziale Netzwerke
  - YouTube
  - MyViedeo, Clipfish, ...
- kommerzielle Dienste
  - Netflix
  - Amazon Video, Hulu, ...
- ...

## **Voice over IP – VoIP**



---

## online-Gaming

## online-Banking

Abwicklung von Bank-Geschäften über's Internet  
heute rund 50 % der Bank-Kunden auch online-Banking-Nutzer

### **Sicherheits-Verfahren für Überweisungen / Anmeldung**

- **TAN / iTAN (TAN-Listen)** Auswahl aus indizierter Kunden-spezifischer Liste  
niedrige Sicherheit  
seit Mitte 2019 nicht mehr möglich
- **TAN-Generator** aus Ziel-Konto-Nummer und Betrag  
hohe Sicherheit
- **smsTAN** separate TAN-Übertragung auf ein Smartphone (als SMS)  
moderate Sicherheit (Gefahr durch geklonte Sim-Karten und gestohlene / verlorene Smartphone's)
- **TAN-Generator mit ...** liefert dynamischen Barcode (mit Kontonummer und Betrag)  
vom Bildschirm und berechnet dann auf dem Gerät eine TAN  
sehr hohe Sicherheit
- **PhotoTAN** farbiges Bild-Punkte-Muster  
sehr hohe Sicherheit
- 

Angriffe richten sich gegen Nutzer, da diese oft nicht genug Wert auf Sicherheit legen  
heute doppelte Authentifizierung (2FA ... Zwei-Faktor-Authentisierung)  
Einloggen mit Kennung und Passwort (Wissen) sowie aktuell erzeugtem TAN (Besitz Generator + Bankkarte)  
alternativ auch biometrische Authentisierung (z.B. Finger-Abdruck, Iris-Scan, ...) möglich  
(Inhärenz)

moderne Zahl-Methoden / online-Bezahl-Dienste (mit virtuellem Zwischen-Konto)

- PayPal
- Apple Pay (nur mit Apple-Geräten möglich)
- google Pay (benötigt Gerät mit NFC-System (Near Field Communication))

Kunde bezahlt nicht mit den sensibelen Kreditkarten-Nummern oder direkter Einwahl bei der Bank sondern "nur" mit den Daten des virtuellem Konto's

---

sofortige Gutschrift beim Verkäufer / Empfänger  
verzögerte Abbuchung beim Kunden / Absender

zusätzlich Abwicklungs-Dienstleister / Direkt-Bezahlung:

- Klarna

ausgelagerte hochspezialisierte Dienste mit hohem Sicherheits-Anspruch  
Direkt-Überweisung

Krypto-Währungen  
mit eigenem Netzwerk (und integrierter Blockchain-Technologie / -Verfahren)  
dezentral, von Banken und Staaten unabhängig  
u.U. keine Zuordnung von Besitzer zu Konto möglich

an Smartphone gebundene Krypto-Währung Libra von google

## *Arbeiten in der Cloud*

zuerst nur als externer Speicher

---

## 8.4.x. IoT – Internet of Things (Internet der Dinge)

Geräte kommunizieren untereinander (auch unabhängig vom menschlichen Nutzer)  
Geräte werden praktisch intelligent (smart)

Industrie 4.0 → Effektivierung und Digitalisierung von Geschäfts-Prozessen

Smarthome  
Gebäude- und Labor-Steuerung  
Intelligente Stromzähler

Sensoren / Aktoren

Smart City  
Ampel-Steuerung in Abhängigkeit des verkehrs-Aufkommen  
Intelligente Stromzähler  
Steuerung der Straßenbeleuchtung

Smart Agriculture  
Überwachung der Umweltbedingungen  
Reife-Überwachung

eHealth  
Fitniss Tracker / Fitness-Armbänder  
Steuerung / Überwachung von Herzschrittmachern

Anforderungen an IoT-Geräte

- hohe Zuverlässigkeit
- geringer Wartungsaufwand
- niedriger energieverbrauch
- geringe Anschaffungskosten

genutzte Technologien (für IoT)

- RFID
- QR-Code
- Barcode
- Sensoren + Aktoren
- div. Internet-Adressierungen und -Protokolle (dazu z.B. MQTT)

8.4.x. Anwendungen / Protokolle der nahen Zukunft

smarte Welt

---

## **8.5. Internet-Sicherheit**

eigentlich als offenes Medium gedacht, in dem alle gleichberechtigt agieren können  
Forschungs-Projekt  
"friedliche" Nutzung durch aufgeklärtes, vertrauenswürdige akademisches Personal  
eingeschränkter Personen.Kreis; praktisch nur Spezialisten  
Hintergedanke militärische Nutzung bestand schon

kriminelle Energie und Potential zu Anfang deutlich unterschätzt  
zuerst ging es um die Funktionalität; Sicherheit war immer zweitrangig  
mehr die Gefahr der unberechtigten (kostenlosen) Nutzung und des Einbruchs in staatliche  
Systeme gesehen  
erst mit der kommerziellen Nutzung des Internet's wurde es auch für wirklich kriminelle Akti-  
vitäten interessant  
Verdienen von Geld möglich, nicht das Erschleichen von Leistungen

heute starke Probleme mit gestohlenen Account's einschließlich diverser Zusatz-Daten (Kre-  
ditkarten-Nummer, Gesundheitsdaten, Profil-Daten, Anschriften, ...)  
z.B. Anfang 2019 großer Leak "Collection #1 - #5" mit über 2,2 Mrd. Zugangsdaten  
damit auch Zugriff auf andere Dienste / Web-Seiten / Plattformen möglich, da Nutzer i.A.  
Passwörter mehrfach nutzen → Angriff auf dieses Verhalten wird "Credential Stuffing" ge-  
nannt

Gefahren auch durch externe Dienste-Anbieter, weil diese nicht alle Anforderungen erfüllen,  
z.T. auch wegen Preisdruck der großen Anbieter / Plattformen

Tummelplatz für Kriminelle / (weitgehend) mögliche Anonymität und Länder-übergreifendes  
Taktieren fördert kriminelle Aktivitäten

Malware-Angriffe auf kritische Systeme (Krankenhäuser, Strom-Versorgung, ...)  
z.B. Verschlüsselung der Daten und dann Erpressung (→ Ransomware)

Das Internet vergisst nicht  
Spuren schwer bzw. niemals löschar  
nichts bleibt unbemerkt

Dark-Net  
geschützte und anonyme Kommunikation  
Handel mit problematischen Dingen (Waffen, Drogen, (Kinder-)Pornographie)

Spionage (Einbruch in Firmen-Netzwerke (vorrangig Entwicklungs-Abteilungen))  
Platzierung von Backdoor's

Fake-News / Desinformations-Kampagnen / ...

social Engineering  
Ausnutzen von typischen menschlichen Handlungsweisen  
Hetze / Hass / Cyber-Mobbing / ...  
Problem der gefühlten absoluten Meinungs-Freiheit  
anonymes / unpersönliches Agieren (Senkung der Hemmschwelle)

Verstoß gegen Urheberrechte / Schutz des geistigen Eigentum

statistische Daten meist positiviert, weil Dunkelziffer bei Meldungen (möglicher Image-  
Verlust))

---

## Was spielt den Kriminellen / ... in die Hände? Ursachen für steigendes Gefahren-Potential

Mrd. von Eintrittsstellen  
viele technisch nur gering gebildete Nutzer  
normal gutgläubige Nutzer

Schwachstellen in Systemen, Protokollen, Programmen, ...  
ist normal  
Design- und Konfigurations-Fehler

Werkzeuge werden immer einfacher zugänglich und benutzbar  
selbst die Veränderung sehr komplexer Viren durch "Baukästen" immer einfacher

fehlende zentrale Organisationen / Kontroll-Organen

Mauschelein der Geheimdienste spielen weitere Rolle

ohne klare Brüche ist aus Forschungs-Projekt eine Schlüsseltechnologie geworden  
Firmen forcieren Entwicklungen, um Funktionalitäten zu erreichen; Sicherheit bleibt zweit-rangig, z.T. bewußt vernachlässigt, weil es Ressourcen bindet

alle vernetzte Geräte sind geeignet

nationale Gesetze und Strafverfolgungs-Systeme gegen international agierende Kriminelle  
(Server z.B. einfach in Länder mit schwachen oder fehlenden Gesetzen und / oder Strafverfolgungs-Organen)  
Internet kennt kaum Grenzen, selbst Geoblocking lässt sich leicht austricksen

Internet ist nicht mehr einfach abschaltbar (auch nicht für einzelne Personen, Institutionen, Länder); lebensnotwendige Infra-Struktur; echtes Gefahren-Potential meist unterschätzt oder klein geredet ("ist ja bis jetzt gut gegangen" sagte der Blinde kurz vorm Rand der Klippe)

extreme Komplexität

hoher Innovations-Druck  
immer schnellere Entwicklungs-Zyklen (praktisch schon so schnell, dass "fertige" Software beim Nutzer (end-)getestet wird (Bananen-Software → reift beim Verbraucher)

die Masse der Anwender ist auf Gefahren nicht vorbereitet!  
Bewußtsein für Informationssicherheit und (Personen-)Datenschutz ist relativ gering  
praktisch nur geringe Rolle in der klassischen Bildung  
Kapitulation vor der Komplexität; jeder ist froh, wenn etwas läuft  
großer Aufwand für Aktualisierung von Software, ...

wenig hochqualifiziertes Personal auch in problematischen Arbeits-Situationen (fehlende Administratoren)  
99 % der Angriffe (auf der Basis von bekannten Schwachstellen) hätten verhindert werden können, weil es meist Konfigurations-Fehler sind oder Update's nicht eingespielt wurden

---

## 8.5.x. Sicherheits-Ziele und Angriffs-Szenarien

### ***Sicherheits-Ziele in Netzen / im Internet***

- **Verfügbarkeit**
- **Vertraulichkeit**
- **Verbindlichkeit**
- **Authentizität**
- **Integrität**
- 

### **unterschiedliches Gefahren, je nach Netzwerk-Typ:**

geschlossene Netzwerke

sind nicht an das Internet angeschlossen

meist auch von anderen (ev. gefährlichen / gefährdeten) Netzwerken abgekoppelt (physikalische Trennung)

Zugang nur durch persönlichen Zugriff oder manipulierter Technik, ...

offene Netzwerke

haben Anschluß zum Internet und / oder anderen Netzen

besonders anfällig, weil praktisch keine Grenzen bestehen

### **Denial of Service-Angriff (DoS-Attacke)**

greift Verfügbarkeit an

Überlastung des Server's etc. durch stark erhöhte Anzahl (meist sinnloser) Anfragen

meist stürzt System / Server ab

### **Verlust der Authentizität / Identitäts-Diebstahl**

Fälschung / Mißbrauch von digitalen Identitäten (→ )

Vortäuschen falscher Identitäten

→ spoofing

### **Integritäts-Verlust / Fälschung von Inhalten / Daten**

veränderte Überweisungs-Beträge beim online-Banking

→

### **Vertraulichkeits-Verlust**

da allgemein das Broadcasting-Prinzip verwendet wird, kann der Datenverkehr mitgehört werden

→ eavesdropping

---

## Verbindlichkeits-Verlust

Problemkreis Rechtsverbindlichkeit / Vertrags-Regeln

### Arten von Angriffen / Attacken

- **aktiv**      Angreifer verfälscht Inhalte, mißbraucht Identitäten, verbraucht zu viele Ressourcen; überlastet Systeme  
spioniert Systeme aus und stiehlt / manipuliert Daten  
Angreifer handelt bewußt  
leichter erkennbar
- **passiv**      Angreifer hört mit, zeichnet Daten-Verkehr auf; liest eMail's und Dokumente mit; spioniert Netzwerke aus (nur "Neugier")  
Angreifer handelt bewußt aber ohne Beeinflussung anderer Nutzer / Systeme  
kaum erkennbar
- durch Fehlbedienungen  
fehlerhafte Daten-Eingaben  
erweiterte Zugriffs-Möglichkeiten durch Konfigurations- und / oder Administrations-Fehler  
...  
"Angreifer" unbewußt / Angreifer wider Willen

Internet Crime Complaint Center (iC<sup>3</sup>)  
für die USA beim FBI angesiedelt  
seit 2000 wurden rund 4,4 Mio. Klagen / Meldungen  
rund 900 Klagen pro Tag



---

### **Risiko-Bereiche**

- **Computer-Systeme** fehlende Backup's  
System-Ausfälle  
Fehler in der Software  
Überlast
- **Netzwerk-Verbindungen** Unterbrechung von Überweisungs-Vorgängen
- **Schnittstellen / Zwischensystem  
in Netzwerken**
- **Nutzer  
human factor** Fehl-Bedienung  
unbeabsichtigtes Löschen / Vernichten von Da-  
ten  
Erschleichen von Wettbewerbs-Vorteilen  
Sabotage  
Mißgunst / Gier / Rache / Langeweile / ...
- **Umwelt** starke Sonnenaktivität (starke Partikel-  
Strahlung)
- **Geheimdienste / Militär**
- ... Schädigung des Image  
Übergang zur digitalen Dokumentation

### **Risiko-Dimensionen**

- **Schadenshöhe**
- **erwartete Häufigkeit**

nach ISO 31000

unabhängig davon auch interessant:

- Anzahl betroffener Systeme
- Arten der betroffenen Systeme
- Aufwand für Schutz / Beseitigung

---

## ***Risiko-Analyse***

- **Phase 1: Definition** der Analyse-Domäne
  - Eingrenzung der betrachteten Bereiche
  -
- **Phase 2: Risiko-Beschreibung** in der Analyse-Domäne
  - Szenario- od. Simulations-basiert
  -
- **Phase 3: Risiko-Bewertung** z.B. nach:
  - Schadenhöhe
  - Eintritts-Wahrscheinlichkeit
- **Phase 4: Ergebnis-Interpretation** Bewertungen und Empfehlungen:
  - Schutz-System
  - Verhaltens-Normen
  - ...

## **online-Banking**

unproblematisch ist i.A. Bestellung / Transaktionen  
meist verschlüsselte Verbindungen  
sicheres TAN-System mit Generator oder Photo-TAN

## **Probleme beim online-Banking**

- unsichere PC's (nicht aktuelles Betriebssystem, fehlende Sicherheits-Software, ...)
- eingeschleuste Software (Trojaner, Daten-Logger, ...)
- 

## **social Networking**

Daten-Übertragung und Speicherung gut gesichert  
Backup-Systeme (system-Ausfälle bleiben meist unbemerkt)

---

## **Probleme beim social Networking**

- ausplaudern von intimen, persönlichen Informationen (u.a. nur aus Geltungs-Bedürfnis oder Unbedachtheit)
- Phishing (z.T. mit Ausnutzung der ausspionierten Personen-Daten)
- (ungeprüfte) Werbung
- Daten-Sammlung
- gläserner Nutzer (Geo-Lokalisation)
- mehrfache Passwort-Nutzung (Erkunden des Passwort's durch gefälschte Seiten)
- indirekte Authentizierung ("mit Facebook anmelden") bei anderen Diensten
- unnötige Kommunikation (sinnfrei Mitteilungen, zu viele Nachrichten, ...)
- Unter-Stress-Setzung / Zeit-Verlust / Angst etwas zu verpassen / sich ständig beobachtet fühlen und ständig verfügbar zu sein / schnelle Umdisponierung
- 

### **Definition(en): Schwachstelle**

Eine Schwachstelle ist eine in einem System vorhandene Möglichkeit, die üblichen (Sicherheits-)Kontrollen und Abwehr-Mechanismen zu umgehen bzw. ein Eindringen / Manipulieren des Systems ermöglichen.

könnte z.B. eine fehlerhafte Programmierung der PDF-Anzeige sein  
die Schwachstelle soll die Ausführung von Remote-Code ermöglichen

### **Definition(en): Exploit**

Ein Exploit ist ein Versuch / eine Aktion zur Ausnutzung einer Schwachstelle.

das passende Exploit zur obigen Schwachstelle wäre jetzt eine präparierte PDF-Datei mit einem ausführbaren Programm, das z.B. weiteren Code nachlädt und dann ausführt  
die PDF wird dann z.B. mit einer eMail verteilt ("Mahnung zu Rechnung 39623")  
dies wäre dann der eigentliche Schadcode

### **Definition(en): Schadcode**

Schadcode ist eine Software innerhalb eines Exploit, mit der eine Schwachstelle genutzt / angegriffen wird.

der Schadcode könnte nun z.B. die Festplatte des "Opfer's" verschlüsseln oder Daten manipulieren  
passiert dies, dann sprechen wir von einem Sicherheitsvorfall

---

## **Definition(en): Sicherheitsvorfall**

Der Sicherheitsvorfall ist der Angriff auf ein System mit Schadwirkung.

heute wird automatisiert nach Schwachstellen gesucht

### ***Angriffsziele***

- **Störung der IT-Sicherheit**
- **Provokation von Abstürzen**      um z.B. beim Neustart eine Default-Start-Situation zu erhalten (von der z.B. die Kennwörter od.ä. bekannt sind)
- **Einsehen / Lesen von Daten**
- **Verschlüsseln von Festplatten**
- **Löschen von Datenbanken**
- **Kopieren von Datenbeständen**      Passwörter, Kreditkartennummern, ...

### ***Klassifizierung von Angriffen nach Ziel-Typ***

- **Einzel-Personen**
- **Unternehmen**
- **Regierungs-Organisationen**
- **IT-Struktur / Netzwerk**

### ***Klassifizierung von Angriffen nach den Absichten (der Angreifer)***

- **Funktionsfähigkeit beeinträchtigen**      z.B. durch Denial of Service-Attacken
- **Diebstahl / Manipulation von Daten**
- **Bloßstellen / Blamieren / Defamieren**      z.B. Defacement
- **Nachweis des Können's**

---

### **Klassifizierung von Angriffen nach dem Aufwand**

- **einfache, schnelle Angriffe** direkte und begrenzte Angriffe
- **erweiterte Angriffe** hoch-komplex, versteckt, verzögert / Zeitpunkt-gesteuert
- **hoch-komplexe Angriffe** hoch-komplex, verschlüsselt, in Phasen aufgeteilt  
z.B. Advanced Persistent Threat's (APTs)

### **Angreifer-Typen / -Arten**

- **Script Kiddies** meist aus Geltungssucht / Neugier / Langeweile /... / Austesten der Grenzen  
verbringen viel Zeit vorm Computer  
Nutzung von fertigen Angriffs-Werkzeugen / Scripten aus dem Internet  
geringe System- und Programmier-Kenntnisse  
geringes Gefühl für IT-Sicherheit und Schadens-Dimensionen
- **Cyberkriminelle** Profitgier steht im Vordergrund / auch Rache  
Empathie für Geschädigte sehr gering  
Rache / viel Gewinn mit wenig Einsatz (ohne eigene Arbeit)  
meist an vielen anzugreifenden Systemen interessiert (mit versteckten / unbekanntem Schwachstellen)  
gekauft Exploit's, spezialisierte Angriffs-Werkzeuge
- **professionelle Hacker** am Aufdecken / Ausspähen von Geheimnissen interessiert  
Wettbewerb mit Anderen (Geltungssucht)  
gute bis sehr gute System- und Programmier-Kenntnisse  
selbstgeschriebene Angriffs-Werkzeuge  
z.T. Arbeiten im Auftrag der (wirtschaftlichen) Konkurrenz
- **staatliche / Staatsnahe Abgreifer** z.B.: CIA, NSA, BND, APT40, Sofacy, Unit 8200, ...  
politisch, militärisch, national-wirtschaftlich motiviert  
Lahmlegung von Einrichtungen anderer Staaten  
Abhören, ..., Erpressen von Entscheidungs-Trägern

<b>Definition(en): Malware</b>
Malware sind Programme.

Kunstwort aus **Malicious Software**,  
beinhaltet viele Arten von bössartiger Software:

---

## Malware-Arten

- **Viren** sind (eingeschleuste) Bestandteile (Miniprogramme) von (ausführbaren) Wirts-Programmen, die für die eigene (lokale) Verbreitung und irgendwelche Schädigungen sorgen
  - Boot-Viren
  - weitere Verbreitung durch Übertragung der ausführbaren Dateien
- **Würmer** hierbei handelt es sich selbstständige Programme, die sich über Netzwerke oder Datenträger (heute vorrangig USB-Stick's) verbreiten Nutzer muss diese Programme nicht starten oder eine Zustimmung dazu geben!  
Hauptschadwirkung ist die Be- bzw. Über-Lastung der Netzwerke  
Nebenschadwirkung (Schadcode (Payload)) kann das gesamte Potential an Schädigungen beinhalten  
häufig Installation von Backdoors (Hintertüren), um selbst nach der Entfernung der Würmer noch eine Zugriffs-Möglichkeit auf den Wirts-Rechner zu haben (solche Rechner können dann ferngesteuert agieren (als Bot in einem Botnet))
- **Trojaner** sind Programmteile in anderen (gewünschten) Programmen (Funktionen), die aber nachteilige / unerwartete Wirkung (Schadwirkung) haben  
schädliche Programmteile sind getarnt / versteckt und warten auf Auslöser
- **Adware** sind Programm, die neben ihrer gewünschten Funktionalität z.B. (nervige) Werbung anzeigen  
auch in Web-Browser möglich; kann dann z.B. alle Werbe-Anzeigen manipulieren bzw. zusätzliche Werbung platzieren oder Nutzer (immer wieder) zu bestimmten Webseiten umleiten
- **Spyware** auch Spionage-Software, Schnüffelsoftware (Sniffer), Spähprogramme, ...  
eigenständige Programme, die ohne Zustimmung des Nutzers Rechner nach sensiblen Daten, digitalen Identitäten, Nutzungsgewohnheiten, Surf-Verhalten, ... durchsuchen bzw. sie erfassen und diese dann an den Angreifer (im Hintergrund und verdeckt) übermitteln  
gerne für personalisierte Werbung / Phishing mißbraucht
- **Key-Logger** kleine eigenständige Programme, die den Tastatur-Puffer mitlesen / mitschneiden (und nach Passwörtern usw. suchen)  
im Rechner als unauffällige Betriebssystem-Dienste getarnt
- **Ransomware** ist ein Programm, dass durch gezielte Aktivitäten die Nutzung des System's unmöglich macht (z.B. Verschlüsseln von Festplatten)  
erpressen Nutzer zur Zahlung von (Bitcoin-)Geld-Beträgen  
ob allerdings eine Dechiffrierung wirklich erfolgt ist fraglich, da hier größere Gefahr für den Angreifer besteht (außerdem hat er sein Geld ja auf einem anonymen Konto)  
meist als Anhänge in eMail's, Trojaner, ...
- **Rootkit**
- ...

---

heute immer mehr hybride Typen, die Funktionalitäten verschiedener Malware-Arten zu neuen – noch gefährlicheren / bösartigeren – Produkten verbinden  
häufig kommt zuerst nur eine kleine "unauffällige" Schadsoftware eines Types, die dann weitere Funktionalitäten nachlädt (ev. Zeit-verzögert, um unauffälliger zu agieren)

Problem Windows / microsoft:

viele Dienste verbinden sich beim Start / immer wieder mit irgendwelchen Servern im Internet, ohne dass deren Funktionalität dies unbedingt erfordern würde

Maßnahmen:

- aktuelle Betriebssysteme (immer aktualisieren)
- professionelle(n) Viren-Scanner / Internet-Security-Suite (kostenfreie Versionen von großen Anbietern bieten nur teilweise Schutz!) → es gibt aber kostenfrei, sehr professionelle Programme und oder Versionen (für Privat-Personen), z.B. von:
  - Avast
  - Comodo
- regelmäßige Sicherung auf einem externen (abkoppelbaren) Datenträger (ev. mit verschlüsselten Ordnern / Zugriffs-Kontrolle)
- regelmäßige externe Kontrolle des eigenen Systems z.B. mittels Live-Systemen (z.B. auf Linux-Basis)
  - z.B. Sicherheits-USB-Stick von
  - auch gute Schutz-Funktion: Virenschanner von portable App's-Systemen
- ...

### **Aufgaben:**

- 1.
2. *Was ist der sogenannte Bundestrojaner? Wo bestehen die Gefahren bei dessen Anwendung!*
3. *Prüfen Sie, wie sich ein direkter Seitenaufruf und die Nutzung eines google-Suchergebnis-Link voneinander unterscheiden!*

### **Beispiele für Angriffs-Szenarien**

Kapern von Accounts (von Prominenten / Konkurrenten / Mitschülern usw.)

lagrer Umgang mit Passwörtern / Passwort-Regeln / ...

Abgreifer meist ohne größere Kenntnisse von den Systemen und / oder der Programmierung

### **erweiterte Angriffe**

Opfer wird vorher (sehr zeit-aufwendig) erkundet (läßt oft den Verdacht der staatlich organisierten Struktur dahinter zu)

Erkennen von Schwachstellen in benutzten System (Webseiten, social-Media, ...)

zusammenstellen einer - ganz genau auf die Person / Organisation zugeschnittene / abgestimmte Schadsoftware

---

Angreifer sind ausgewählter, hoch qualifizierter Personen-Kreis mit guten bis sehr guten technischen und Programmier-Kenntnissen

## **Phishing**

Angreifer verschicken (täuschend echt aussehende) Mail's von Banken, online-Shop's usw. usf. um durch irgendwelche Tricks / Geschichten an die Anmelde-Daten der Kunden zu kommen

gute Prophylaxe:

- Absender prüfen (exakte Schreibung der Adresse, Prüfen des Link's durch Maus-Drüber)
- Inhalt / Sachverhalt prüfen
  - Banken erfragen nicht per eMail Nutzerdaten
  - Rechtsanwälte verschicken Schriftsätze nicht per eMail
  - ...
- Rechtschreibung und Zeichensätze prüfen
- unbekannte Anhänge nicht öffnen
- Trennung von Postfächern nach Zweck (echte Kommunikation (Firma, Freunde) und Anmeldung bei online-Diensten)
- regelmäßiges Ändern der Passworts
- ...

gute Abwehr-Techniken:

- aktuelles Betriebssystem
- aktueller Viren-Scanner (besser Internet-Schutz-Suite)
- Anwender-Programme aktualisieren / updaten
- Gehirn-Computer einschalten!!! (Neugier und Habgier sind Freunde der Internet-Kriminellen)
- lieber erst noch einmal bei angeblichen Absender nachfragen (sensibilisiert auch den angeblichen Absender!)
- Spam-Filter einschalten / trainieren
- bei aufgerufenen Seiten falschen Account und irgendetwas sinnloses als Passwort eingeben, wenn hier keine Fehlermeldung kommt, dann ist sehr wahrscheinlich was faul → Verdacht der betroffenen Website melden
- unabhängige Anmeldungen bei den einzelnen Diensten (z.B. Dienstname in verkürzter Form mit ins Passwort integrieren)
- gute Passwörter verwenden
- mit isolierten Systemen arbeiten (VirtualBox od.ä.; SandBox-Systeme, ...); Kombination von Linux- und Windows-Betriebssystem als Host und virtuelles System
- Kiosk-Modus (God-Modus) verwenden (verhindert Schreiben in registry und / oder auf Datenträger)
- Backup's / Systemzustände speichern
- ungewöhnliche / abweichende Einstellungen vornehmen (Schadsoftware ist auf typische Systeme eingestellt)
- ...

Bei den Banken ab 2019 verschärfte Anmelderegeln

2-Faktor-Authorisierung (Kombination von zwei unterschiedlichen Sicherheit-Kennzeichen (z.B. Passwort und frisch generierte TAN))



---

## **Spear Phishing**

richtet sich an engen Personenkreis oder Einzel-Personen  
mit hohem Vorbereitungs-Aufwand

## **Malware ILOVEYOU**

stammt aus Philippinen (ab 2000)

EMail mit VBS-Anhang (VisualBasic-Skript) getarnt als "Love-Letter-for-you.txt" hatte zudem den Betreff "ILOVEYOU"

VBS-Datei-Typ war wegen der Ausblendbarkeit in Windows-Systemen nicht sichtbar, VBS sind Skript-Dateien, die auf Betriebssystem-Ebene arbeiten können (mit den Nutzer-Berechtigungen), viele Nutzer sind traditionell (nur ein Konto pro Rechner) und aus Bequemlichkeit mit einem Administrator-Account aktiv

nach der Aktivierung des Skript's verbreitete sich das Programm (praktisch wie ein Wurm) an alle Kontakte im Windows-Adressbuch

da die versendeten Mail von vermeintlich bekannten Kontakten stammte, wurde sie bei den Empfängern geöffnet und konnte sich dann exponentiell verbreiten

geschätzter Schaden zwischen 5,5 und rund 9 Mrd. Dollar

10 % aller Internet-Nutzer betroffen (in den ersten 10 Tagen kam es zu mindestens 50 Mio. Infektionen)

## **WannaCry-Kampagne**

2017; mehr als 200'000 betroffene Computer

nutzt geleakte Schadsoftware (Eternal Blue) der NSA und greift veraltete Windows-Systeme und –Versionen an

Daten werden verschlüsselt, Entschlüsselung nur nach Zahlung (Bitcoin)

Verschlüsselung wurde von geknackt und passendes Entschlüsselungs-Tool bereitgestellt

höchst-spezialisierte Angriffe (APT (Advanced ))

extrem aufwändig – auch durch Verschleierung der Herkunft des Angriffs (höchst-spezialisiertes Personal, große Zeit-Ressourcen, viel Vorfeld-Arbeit / Grundlagen-Forschung, große Rechner-Kapazitäten, ...)

praktisch nur von staatlichen / Staats-nahen Organisationen leistbar (Geheimdienste, Hacker-Gruppen, ...)

Ausnutzung sehr spezieller und versteckter / geheimgehaltener Schwachstellen (Zero-Day-Schwachstellen)

gerade bekannt gewordene Schwachstellen genutzt, für die aber noch keine Patches vorliegen (→ Zero-Day-Schwachstellen),

Ziele praktisch immer politisch oder wirtschaftlich

- Manipulation von Wahlen
- Störung fremder Infrastrukturen (Energie, Telefon, Internet, ...)
- Öffentlich-Machung von illegalen Projekten, ...
- Informations-Beschaffung

## **Stuxnet**

berühmtes Beispiel; Wurm greift Siemens-Steuerungs-Systeme (SCADA-Kontroll-Software) an, wahrscheinlich entwickelt vom israelischen Geheimdienst, gelangte durch social Hacking

---

(Ausnutzung menschlicher Schwächen) in iranische Uran-Anreicherungs-Anlagen und ließ dort die Zentrifugen zu schnell drehen, bis zum Defekt  
weitere Schwachstellen in Windoes-Betriebssystemen und –Netzwerken ausgenutzt schon 2007 im Umlauf gebracht, aber erst 2010 entdeckt / bekannt geworden  
das Netz der Uran-Aufbereitungs-Anlage war vollständig isoliert, Einbruch durch einen USB-Stick, den ein Mitarbeiter mitgebracht hatte und unberechtigt angeschlossen hat

### **Emotet – Banking-Trojaner**

2014 entdeckt

wurde entwickelt, um Accountdaten zum online-Banking zu stehlen

später durch Spam-Funktionen erweitert

Verbreitung über gut gefälschte Spam-Mail's

nach Infektion wird Kontakt-Liste zur Weiterverbreitung genutzt

noch 2019 werden Infektionen von Emotet gemeldet!

### **Zero-Day-Angriff**

Es werden hierbei frisch erkannte Schwachstellen genutzt. Meist hatten die Software-Hersteller oder Dienste-Anbieter noch keine Zeit, die Schwachstelle zu beheben und ein Patch etc. bereitzustellen.

Einzig effektive Gegenmaßnahme ist hier die Meidung / Nichtbenutzung der betroffenen Software bzw. des Dienstes. Ev. sollte auch eine weitgehende Internet-Abkopplung in Betracht gezogen werden.

---

## 8.5.x. digitale Identität

Einrichten mit "Registrieren"

Einrichten des Internet's beim Internet-Service-Provider

Da im Internet niemand physisch aktiv sein kann, muss als Äquivalent eine digitale Identität genutzt werden.

notwendig für:

- Herstellung der Situation, wie eine Web-Site verlassen wurde (→ z.B. Cookies)
- Adresse und Konto-Verbindung für online-Einkäufe
- herunterladen von Software / Musik / ...
- Nutzungs-Daten (Verbrauchs-Abrechnung)
- ...

### **Definition(en): digitale Identität**

Eine digitale Identität ist eine Sammlung von Daten zur Charakterisierung einer physischen Identität im Internet oder in einem Software-System.

Haupt-Eigenschaften sind:

- eMail-Adresse
- Account-Name / Nickname
- Passwort
- ...

i.A. obligatorisch (in wechselnder Zusammensetzung und Anzahl)

Minimal-Angaben

zugeordnete Neben-Eigenschaften sind:

- Konto-Verbindung
- Kreditkarten-Nummer
- Telefon-Nummer
- Adresse
- ...

diese Angaben sind meist fakultativ / optional  
für die normale Nutzung i.A. notwendig

nach Registrierung ist die Anmeldung beim Dienst usw. über die "Anmeldung" möglich  
hier erfolgt Zugriffs-Kontrolle durch den Dienst, Prüfung der digitalen Identität  
Prozess ist die Authentifizierung, was praktisch bedeutet, dass eine physische Identität (sich anmeldender Nutzer) mit einer (vorher registrierten) digitalen Identität verknüpft wird

---

## Definition(en): Authentifizierung

Ist das Verfahren zur Prüfung einer Verbindung von einer physischen Identität mit einer (registrierten) digitalen.

Anforderungen an die Authentifizierung:

- sicher (zweifelsfreie Passung von physischer und digitaler Identität)
- geschützt vor:
  - Diebstahl
  - Mißbrauch
- ...

mögliche Zugriffs-Kontrollen / Authentifizierungs-Verfahren:

- Nutzernamen / eMail-Adresse - Passwort (Wissen)
  - nicht für Internet und seine Anforderungen entwickelt, aber wegen der Einfachheit übernommen
  - sehr weit verbreitet
  - recht unsicher (möglicher Identitäts-Diebstahl)
    - angreifbar durch Mitlesen von Betätigungs-Mails, Mails wegen Passwort-Vergessens
    - Phishing
    - Erraten / Ausprobieren / Wörterbuch-Suche / ...
    - Nutzung von Leak's
    - social Engineering
    - Key-Logger
    - Man-in-the-Middle-Attacke
    - ...
  - ...
- ...

### Problemkreis Identitäts-Diebstahl

Nutzung / Mißbrauch der Ressourcen im Namen der gehakten digitalen Identität

Dieb erhält Zugriff auf die Ressourcen (er ist autorisiert diese zu nutzen)

oft ist es für den regulären Besitzer schwierig / aufwändig sich anzumelden / Mißbrauch nachzuweisen / den Dieb zu identifizieren

Diebstahl ist wegen des schweren Nachweises sehr interessant

## Definition(en): Autorisation

Unter Autorisation versteht man die Freigabe von Rechten / Ressourcen für eine digitale Identität..

Angriffe durch Diebstahl der Kunden-Datenbank

Phishing

Malware / Spyware

Key-Logger

---

## Redseeligkeit in sozialen Netzwerke und / oder Foren

Prüfen, ob digitale Identität geleakt wurde z.B. über folgende Adressen:

- <https://sec.hpi.de> (HPI Identity Leak Checker)
- 

Reaktionen bei geleakten Account's:

- unbedingt sofort Passwort bei allen Diensten usw. deutlich verändern! auch gegeneinander
- Verwendung sicherer Passwörter
- weitere Accounts mit ähnlichen Anmeldungen (benutzten Passwörtern) ebenfalls unbedingt ändern!

## 8.5.x. Verschlüsselung im Internet

### Vorteile:

- erhöhte Sicherheit, Vertraulichkeit, Verbindlichkeit, Integrität, ...
- verhindert Probleme bei Authentifizierung
- Verhinderung diverse Angriffs-Szenarien
- ...

### Nachteile:

- erhöhter System- und persönlicher Aufwand
- Anfällig gegen Fehler bei Installation und Konfiguration
- ...

### Definition(en): Kryptographie

Kryptographie ist die Wissenschaft von der Verschlüsselung und (der regulären) Entschlüsselung von Nachrichten / Daten.

ist stark mit der Informatik und heute auch mit der Mathematik verbunden  
z.B. Nachweis führen, dass ein Verfahren sicher / sehr unwahrscheinlich zu knacken ist  
Entwickeln von Ein-Weg-Funktionen  
(statistische) Analyse von verschlüsselt Nachrichten und / oder kryptographischen Verfahren, um diese zu knacken

### Grund-Modell der Kommunikation



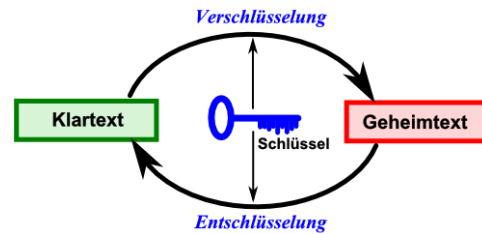
Codierung als (notwendige) Zwischenstufe

eigentlich sind Verfahren und Code-Tabelle bekannt; vorrangig aus technischen Gründen gewählt  
 eigentliche Nachricht und Übertragungs-Medium sind nicht kompatibel / aufeinander abgestimmt  
 bei Unkenntnis wird Codierung bei Mitleser als Chiffrierung empfunden  
 meist sehr leicht knackbar / dechiffrierbar



### Modell der symmetrischen Verschlüsselung

auch Secret-Key-Cryptography  
 zum Verschlüsseln und Entschlüsseln wird der gleiche Schlüssel und praktisch auch das gleiche Verfahren (ev. die Umkehrung) verwendet



Problem des Schlüssel-Tausch, der Schlüssel muss unbedingt geheim gehalten werden  
 historisch sehr alt (mindestens bis zu CÄSAR () zurück (CÄSAR-Verschlüsselung (CÄSAR-Chiffre)  
 allgemein sind Permutations- und Ersetzungs-Verfahren  
 CÄSAR-Chiffre ist Ersetzungs-Verfahren

#### Vorteile:

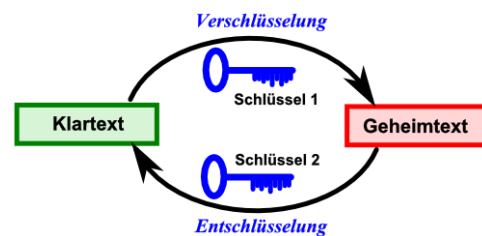
- schnell
- bei Einmal-Schlüsseln (mit mindestens Nachrichten-Länge) praktisch nicht knackbar
- 

#### Nachteile:

- durch notwendigen Schlüssel-Tausch unsicher
- Sender und Empfänger müssen und dürfen ausschließlich den Schlüssel kennen

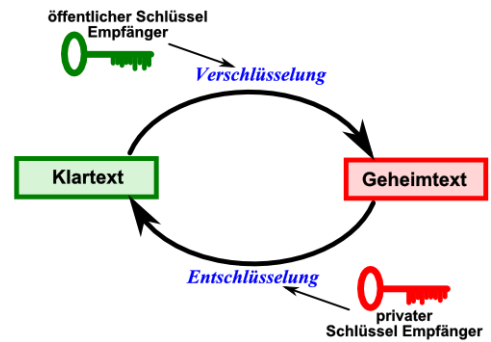
### asymmetrische Verschlüsselung

auch Public-Key-Cryptography  
 für Verschlüsselung und Entschlüsselung werden zwei unterschiedliche Schlüssel und / oder Verfahren benutzt



es gibt für jeden Nutzer 2 Schlüssel, die aber mathematisch ein Paar bilden  
 Verbindung aber nicht zurückzuverfolgen (Einweg-Funktionen; schwer lösbare mathematische Probleme)  
 erst im 20. Jhd. entwickelt (für Internet-Kommunikation)

Im Public-Key-Verfahren wird zum Verschlüsseln einer Nachricht der öffentliche Schlüssel verwendet. Nur mit Hilfe des – geheim gehaltenen - privaten Schlüssels lässt sich die Nachricht wieder dechiffrieren. Problem des Schlüssel-Tausch entfällt, da für jeden Nutzer ein öffentlicher Schlüssel bekannt ist.



Vorteile:

- sehr sicher
- 

Nachteile:

- aufwendiges Vorverfahren (Schlüssel-Erzeugung / -Tausch (Organisation))
- Sicherheit des öffentlichen Schlüssels ist nicht garantiert (Trust-Problem und Man-in-the-Middle-Angriff)
- sehr Rechen- / Ressourcen-aufwendig
- es können gefälschte Nachrichten (mit dem öffentlichen Schlüssel) erzeugt werden (Sender nicht prüfbar)

hybride Verfahren nutzen asymmetrische Verschlüsselung zum Schlüssel-Tausch (für ein symmetrisches Verfahren)  
dann Nutzung des schnellen symmetrischen Verfahren's für eine Sitzung / Verbindung / Nachricht / ...

<b>Definition(en): Kryptologie</b>

### Absicherung der Herkunft einer Nachricht durch Signieren

praktisch das Unterzeichnen der Nachricht

Signieren der übermittelten Nachrichten mit dem privaten Schlüssel des Senders  
Signatur enthält Absender-Angaben und z.B. eine Prüfsumme über die Nachricht  
Übertragung von Nachricht (diese kann und sollte natürlich verschlüsselt sein, für die Signatur ist die Verschlüsselung uninteressant)  
Empfänger kann Signatur mit Hilfe des öffentlichen Schlüssels des Senders prüfen / entschlüsseln, nur dann erhält er die (in der Signatur enthaltenen) Absender-Informationen

Lösung des Trust-Problem's durch Zertifikate

ausgestellt von Trust Center

---

Zertifikat enthält in verschlüsselter Form (mit privatem Schlüssel des Trust Center chiffriert) die Identität eines Nutzers und dessen öffentlichen Schlüssel sowie technische Informationen, wie z.B. Informationen zum Zertifikats-Aussteller und ein Ablaufdatum  
Zertifikate sind öffentlich (haben praktisch Ausweis-Charakter)  
das Zertifikat kann mit dem öffentliche Schlüssel des Zertifikats-Aussteller geprüft werden

### Ablauf einer verschlüsselten Kommunikation im Internet (mittels hybrider Verschlüsselung)

#### **Vorlauf**

- beide Kommunikations-Partner vereinbaren ein **symmetrisches Verfahren X**
- beide Kommunikations-Partner vereinbaren ein asymmetrisches Verfahren Y
- beide Kommunikations-Partner generieren ihr Schlüssel-Paar  $S_Y[PS_Y, ÖS_Y]$  für das Verfahren Y

#### **Schlüssel-Austausch**

- Partner1 (Sender): Generieren eines geheimen **Sitzungs-Schlüssel  $S_x$**  für das symmetrische Verfahren X
- Partner1: Verschlüsseln des Sitzungs-Schlüssel  $S_x$  mit dem **asymmetrischen Verfahren Y** und dem **öffentlichen Schlüssel  $ÖS_Y[Partner2]$**  und schickt das Ergebnis (Sitzungs-Schlüssel als Geheimtext) an Partner2 (Empfänger)
- Partner2: Entschlüsseln der übertragenen Nachricht mit dem **asymmetrischen Verfahren Y** unter Verwendung des eigenen **privaten Schlüssel  $PS_Y[Partner2]$**  (ergibt Sitzungs-Schlüssel  $S_x$  im Klartext)

#### **eigentliche Kommunikation**

- Partner1 und 2 benutzen ab nun das **symmetrische Verfahren X** mit dem Sitzungs-Schlüssel  $S_x$

#### **Kommunikations-Ende**

- Verwerfen des Sitzungs-Schlüssel  $S_x$

### Authentifikation mit Zertifikaten

- Sender erstellt eine Nachricht (egal, ob dies eine lesbare oder verschlüsselte Datei ist)
- Sender signiert die Nachricht (Sender verschlüsselt die Nachricht mit seinem privatem Schlüssel)
- Sender übermittelt signierte (verschüsselte) Nachricht und sein eigenes Zertifikat an Empfänger
- Empfänger prüft / validiert das Zertifikat bei einem Trust-Center (Empfänger kann mit dem öffentlichen Schlüssel des Trust-Center das Zertifikat entschlüsseln)
- mit dem im entschlüsselten Zertifikat enthaltenen öffentlichen Schlüssel vom Sender kann nun die Signatur geprüft werden (Empfänger entschlüsselt mit dem öffentlichen Schlüssel aus dem (entschlüsselten) Zertifikat die signierte (verschüsselte) Nachricht)

### Anwendung der Verschlüsselungs-Techniken beim HTTPS

Secure-Version zum HTTP

es wird eine Zusatz-Schicht zwischen Transport- und Anwendungs-Schicht

das Verfahren ist TLS () (hervorgegangen aus dem SSL von Netscape)

TLS ermöglicht sichere gegenseitige Authentifizierung mit Zertifikaten



---

Ablauf nach dem hybriden Verschlüsselungs-Prinzip

- zuerst TLS-Handshake zur Vereinbarung der verwendeten **symmetrischen** und **asymmetrischen** Verfahren **X** und **Y**
- Browser (Client, Anfrager) sendet "Hello" an Web-Server (Anbieter)
- Web-Server antwort mit "Hello"
- Web-Server sendet sein Zertifikat an den Browser
- Browser verifiziert Zertifikat über das Trust-Center und holt sich den öffentlichen Schlüssel des Web-Server's heraus
- erstellt ein "Pre-Master-Secret" und verschlüsselt dies mit dem extrahiertem öffentlichen Schlüssel (zum **asymmetrischen** Verfahren **Y**) und sendet es an den Web-Server
- Web-Server kann jetzt mit seinem eigenen privaten Schlüssel das "Pre-Master-Secret" entschlüsseln (mit **asymmetrischen** Verfahren **Y**)
- Web-Server und Browser berechnen unabhängig voneinander aus dem "Pre-Master-Secret" den Sitzungs-Schlüssel für die weitere Kommunikation (mittels **symmetrischer** Verschlüsselung)
- Kommunikation mit **symmetrischer** Verschlüsselung bis Sitzungs-Ende

#### **KERCKHOFFSche Regel / KERCKHOFF's Prinzip**

**Die Sicherheit eines Systems darf nicht von der Geheimhaltung des Algorithmus abhängen, sondern nur von der Qualität / Geheimhaltung des Schlüssel's.**

Nachrichten / Daten verstecken ist aber trotzdem eine effektive Methode (verschlüsselte) Nachrichten und / oder auch Schlüssel zu transportieren

#### **Definition(en): Steganographie**

Steganographie ist die Wissenschaft von / sind die Verarbeitetechniken zum (unauffälligem) Verstecken von Nachrichten / Daten in anderen Nachrichten / Daten.

---

## 8.5.x. allgemeine Sicherheits-Empfehlungen

### *sicheres Passwort*

- mindestens 12 Zeichen lang
- mindestens 2 der nachfolgenden Vorschriften einhalten
  - Klein- und Groß-Buchstaben verwenden
  - Ziffern verwenden
  - Sonderzeichen verwenden
- keine "Wörterbuch-Worte"
- für jeden Dienst ein individuelles Passwort festlegen
- Passwort spätestens nach halben Jahr wechseln
- keine Namen, genutzte Artikel / Waren (z.B. Lieblings-Schokolade, ...), nie Geburtsdatum oder Heiratsdatum, Wohnort, ...
- ...

Empfohlen werden Passwort-Manager (möglichst auf einem USB-Stick und / oder online)  
Verwendung eines einzigen supergutem Passwort  
können dann auch zufällig erzeugte Passworte verwenden und z.T. auch automatisch bei einer Anmeldung auf einer Web-Site etc. einfügen

Beispiele:

- KeePass
- 1Password
- LastPass

### *2-Faktor-Authentifizierung*

Kombination von 2 der 3 Authentifizierungs-Methoden (Faktoren)

- Wissen (z.B. Passwort)
- Besitz (z.B. Bank-Karte mit TAN-Generator)
- biometrische Merkmale (z.B. Finger-Abdruck)

Multiplikation der Sicherheiten

### *Datenträger-Verschlüsselung*

Daten sind das Gold des 21. Jahrhunderts

z.B. bei mobilen Geräten (Laptop's) unbedingt zu empfehlen (Diebstahl möglich)

z.B.:

- BitLocker (windows)
- FileVault (apple)

- 
- VeraCrypt

### **Prinzip der Daten-Sparsamkeit**

Welche Daten braucht ein Service etc. wirklich? Need-toKnow-Prinzip

Hier kann man mal die halbe Wahrheit sagen oder irgendwelchen "Quatsch" eintragen! Auch leicht fehlerhafte Einträge wirken Wunder. Ev. kann man so auch ein Daten-Leck (Leak) erkennen, wenn man seinen echten Namen - so ganz ausversehen – mit langen / doppelten Buchstaben eingibt.

Bei Vorname kann man auch auf eine Anrede zurückgreifen, usw. usf.

Sachlich ist das nicht ganz ok, aber die Anbieter sind auch verpflichtet (Datenschutz-Grundverordnung) nur die notwendigen Daten zu erfassen. Scheinbar halten sie sich nicht ganz genau daran. Also: Did-for-dat!

Aber Achtung, Falsch-Angaben können auch rechtliche Folgen haben. Hier muss man genau für sich abwägen. Ich glaube auch nicht, dass ich ein Anwalts-Schriftsatz an meine ausgedachte Adresse bei meinem Liebling-Download-Portal bekomme (-:-).

### **Aufgaben:**

- 1.
2. ***Immer am 01. Januar bekomme ich ganz viele Geburtstags-Grüße, obwohl ich da gar nicht Geburtstag habe. Was ist denn da schief gelaufen?***
- 3.

Auch für App's auf dem Smartphone ist eine genaue Überprüfung der gewünschten Daten-Zugriffe zu empfehlen. Ich habe bis heute nicht verstanden, wozu meine Taschenlampen-App eigentlich meinen GPS-Standort oder meine Einkäufe wissen muss.

### **Zurückhaltung in sozialen Medien**

viele können mitlesen

wenn man hier schreibt, dass man gerade in der Türkei Urlaub macht, dann muss man sich nicht wundern, wenn sich irgendwelche Personen in der Wohnung umsehen, die man Foto präsentiert hat

### **Updates bei Programmen**

erste Viren-Scanner und Internet-Schutz-Suiten bieten schon die Möglichkeit die Aktualität von Programmen zu prüfen und diese ev. auch gleich zu aktualisieren  
es geht vorrangig um das Schließen von Sicherheits-Lücken / Schwachstellen  
die vorgeblich neuen Funktionen sind nur Lock-Mittel und sollen über den eigentlichen Update-Zweck hinwegtäuschen

ehrlicher Umgang mit Fehlern ist übrigens mehr Vertrauens-bildend (siehe Linux-Community)

---

ältere Programme möglichst meiden, da Angreifern die Schwachstellen bekannt sind und oft so alte Programme nicht mehr betreut werden  
alternativ kann man für den Zeitraum der Arbeit die Netzwerk-Verbindungen unterbrechen

### **Updates für das Betriebssystem**

unbedingt notwendig  
nicht mehr betreute Versionen sollten unbedingt gemieden werden  
gerade das Nutzen von Sicherheits-kritischer Software (z.B. online-Banking) wird jetzt zum Russisch-Roulette

mittlerweile schiebt auch windows sehr regelmäßig Updates nach

### **Anti-Viren-Software / Internet-Security-Suiten**

Name ist veraltet, heute müssten sie eigentlich Anti-Malware-Programme heißen  
ein Muss für jeden Internet-Rechner oder Rechner in einem Netzwerk  
überwachen Laufwerke und die darauf enthaltenen Dateien, den Systemspeicher, das Betriebssystem und die Netzwerk-Verbindungen  
gefährliche Programme / Dateien werden in eine Quarantäne verschoben  
da die Viren-Scanner nach bestimmten Code-Schnipseln oder verhaltensweisen suchen, sind Fehler-Meldungen nicht vermeiden. Die meisten Programme bieten einen Lern-Modus. Der ist zu Anfang etwas nervig, weil er sich ständig bei neuen "Problemen" meldet, dafür ist er später umso besser an die eigenen Bedürfnisse angepasst. Man sollte keine Angst vor dem Abschalten haben, die Aktionen können immer wieder zurückgenommen werden. Wenn auf einmal nicht's mehr geht, dann schaltet man die Funktionen oder Programme eben wieder zu.

Meldungen von Viren-Programmen immer ernst nehmen und im Zweifel eine Profi fragen.

empfohlen werden die Voll-Versionen, die kosten zwar, bis auf wenige Ausnahmen Geld, aber bei Kauf von mehreren Lizenzen für z.B. zwei Jahre werden die Preise mehr als moderat

oft gibt es auch auf den DVD's, die in Computer-Zeitschriften stecken, Jahres-Lizenzen für einen Rechner

die einfachen "Test"-Versionen bieten einen akzeptablen Schutz, sind aber nicht ausreichend für Rechner mit sensiblen Daten und Programmen

früher wurde der Virenschutz durch microsoft immer belächelt, mittlerweile ist es als minimale Sicherheits-Stufe akzeptabel

### **externe Antiviren-Programme**

viele Linux-Systeme bieten die Möglichkeit von einer CD/DVD bzw. einem USB-Stick zu booten. Diese Live-Systeme eignen sich sehr gut zum Untersuchen und Reparieren von Windows-Laufwerken. Da windows dann nicht startet, werden auch viele versteckte Programme nicht aktiv (die vielleicht Tarn-Mechanismen aktivieren).

---

Z.B gibt es von SARDU (→ ) ein USB-Menü-System zum Starten von mehreren Live-Systemen mit vorinstallierten Virenschaltern.  
Auch solche USB-Menü-Systeme für portable App's oder ??? enthalten auch Virenschalter.  
Hierfür muss allerdings windows schon laufen.

## **Backup's / Datensicherungen**

je nach Datenbestand täglich bis wöchentlich  
vor allem die Daten-Verzeichnisse / -Laufwerke  
möglichst auf externen Daten-Trägern, die sonst abgekoppelt werden

super Programm, das praktisch alles kann, ist PersonalBackup von RATHLEV  
gibt es kostenlos im Internet, eine Spende ist immer gern gesehen und hat das Programm wirklich verdient  
z.B. Zeitpläne realisierbar; Sichern auf verschiedensten Datenträgern oder Cloud's und auch über FTP  
mit oder ohne Komprimierung und / oder Verschlüsselung

## **Firewall**

gibt es als Hardware und Software  
Hardware eher für den professionellen Bereich (Firmen, Büro's, Praxen, ...), überwachen das gesamte Netzwerk  
Software z.B. direkt mit dem Betriebssystem ausgeliefert und meist auch aktiviert (z.B. windows heute) (praktisch ausreichend, aber eben als windows-Standard-Programm eher für wirksame Schwachstellen anfällig); Software-Firewall's sind fast ausschließlich nur auf dem lokalen System wirksam  
Meldung beachten, vor allem bei Wiederholungen und neuartigen Häufungen  
legen i.A. Protokolle in Text-Form (Log-Dateien) an, die jedermann einsehen kann  
oder als extra Programm, vielfach in den Internet-Security-Suiten enthalten  
überwachen Netzwerk-Verkehr und gestatten / blockieren Verbindungen (z.B. über bestimmte Port's)

unter Linux ist das iptables

---

## Literatur und Quellen:

- /1/ ISBN
- /1/ ISBN
- /3/ KADERALI, Firoz:  
Kommunikationsnetze und –protokolle – Offene Systeme, X.21 und X.25 Protokolle, LANs und MANs, ISDN, B-ISDN, GSM, WLAN, Internet Protokoll, Agenten im Internet  
[www.kalderali.de](http://www.kalderali.de)
- /4/ DAUSCH, Martin:  
Netzwerke – Grundlagen.-HERDT-Verl.-8. Ausgabe, 1. Aktualisierung, Juli 2013
- /5/ DEMBOWSKI, Klaus:  
Computernetzwerke – Der leichte Einstieg in Grundlagen und Praxis.-München, Boston, Sa Francisco, Harlow, Don Mills, Sydney, Mexico City, Madrid, Amsterdam: Addison-Wesley Verl.; 2012  
ISBN 978-3-8273-3092-5
- /6/ BÖHM, Christian; KRÖGER, Peer:  
Einführung in die Informatik: Systeme und Anwendungen – Kapitel 4: Rechnernetze.- Skript zur Vorlesung Sommersemester 2008  
<http://www.dbs.ifi.lmu.de/Lehre/InfoNF>
- /1/ ISBN
- /1/ ISBN
- /A/ Wikipedia  
<http://de.wikipedia.org>

Die originalen sowie detailliertere bibliographische Angaben zu den meisten Literaturquellen sind im Internet unter <http://dnb.ddb.de> zu finden.

---

**Abbildungen und Skizzen entstammen den folgende ClipArt-Sammlungen:**

/A/ 29.000 Mega ClipArts; NBG EDV Handels- und Verlags AG; 1997

/B/

andere Quellen sind direkt angegeben.

**Alle anderen Abbildungen sind geistiges Eigentum:**

// lern-soft-projekt: drews (c,p) 1997 – 2023 lsp: dre

*verwendete freie Software:*

- **Inkscape** von: inkscape.org ([www.inkscape.org](http://www.inkscape.org))
- **CmapTools** von: Institute for Human and Maschine Cognition ([www.ihmc.us](http://www.ihmc.us))
- **Filius** von: Dr. St. FREISCHLAD (<http://www.lernsoftware-filius.de>)
- 

⌘- (c,p)2015 - 2023 lern-soft-projekt: drews -⌘  
⌘- [drews@lern-soft-projekt.de](mailto:drews@lern-soft-projekt.de) -⌘  
⌘- <http://www.lern-soft-projekt.de> -⌘  
⌘- 18069 Rostock; Luise-Otto-Peters-Ring 25 -⌘  
⌘- Tel/AB (0381) 760 12 18 FAX 760 12 11 -⌘