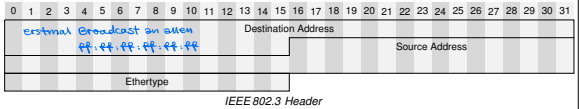
Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

L2: MAC-Address of:fe:44:of:fe:20 → Hardware Address
L3: IP-Address 192.168.1.1/24 → Protokoll Address

Formelsammlung zu Grundlagen Rechnernetze und Verteilte Systeme (GRNVS IN0010)

SS 2022, Seite 1/2, Version: 2022-05-17

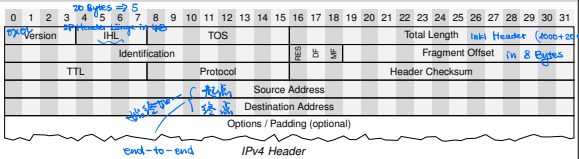
Ethernet / IEEE 802.3 Header und Ethertypes



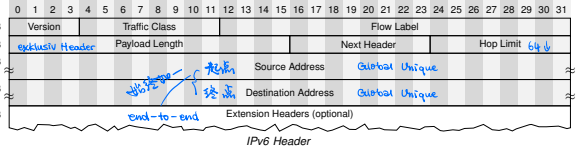
IEEE 802.3 Header

Ethertype	Keyword	Protocol
0x0800	IPv4	Internet Protocol, Version 4
0x0806	ARP	Address Resolution Protocol
0x0842	WoL	Wake-on-LAN Magic Packet
0x8035	RARP	Reverse Address Resolution Protocol
0x814c	SNMP	Simple Network Management Protocol
0x86dd	IPv6	Internet Protocol, Version 6

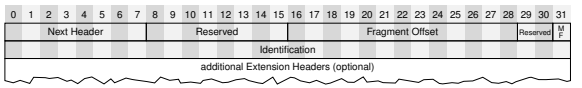
IPv4/6 Header und IP Protocol Numbers bzw. Next Header



IPv4 Header



IPv6 Header

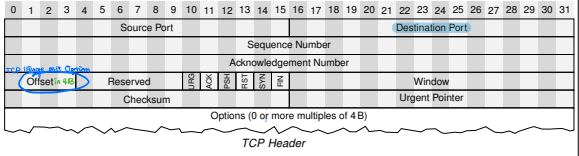


IPv6 Fragmentation Header

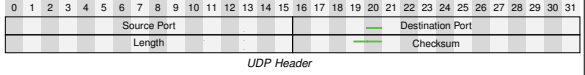
No / NH	Protocol
0x01	ICMPv4 (Internet Control Management)
0x06	TCP (Transmission Control)
0x11	UDP (User Datagram)
0x2c	Fragment Header
0x2f	GRE (General Routing Encapsulation)
0x3a	ICMPv6 (ICMP for IPv6)
0x3b	No Next Header
0x84	SCTP (Stream Control Transmission)

TCP/UDP Header und ausgewählte well-known Ports

für Port 1000 < x < 64000



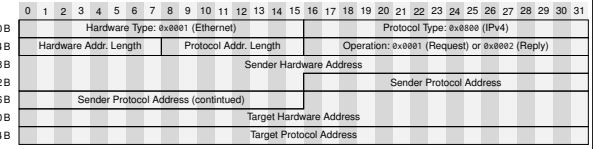
TCP Header



UDP Header

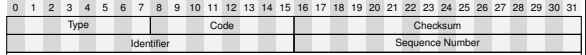
Port	Service Name	Port	Service Name
20/21	ftp	68	bootpc
22	ssh	80	http
23	telnet	110	pop3
25	smtp	443	https
53	domain (dns)	546	dhcpv6-client
67	bootps	547	dhcpv6-server

ARP

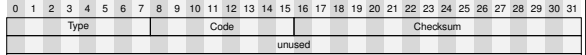


ARP Packet Format

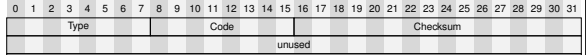
ICMPv4



ICMPv4 Echo Request/Reply



ICMPv4 Time Exceeded



ICMPv4 Destination Unreachable

Type	Code	Description
0 – Echo Reply	0	Echo reply
1 and 2		*Reserved*
3 – Destination Unreachable	0	Destination network unreachable
	1	Destination host unreachable
	2	Destination protocol unreachable
	3	Destination port unreachable
4 – Source Quench	0	Source quench (congestion control)
5 – Redirect Message	0	Redirect Datagram for the Network
	1	Redirect Datagram for the Host
8 – Echo Request	0	Echo request
11 – Time Exceeded	0	TTL expired in transit
	1	Fragment Reassembly Time Exceeded

Ausgewählte ICMPv4 Types/Codes

Zahlensysteme 1/2

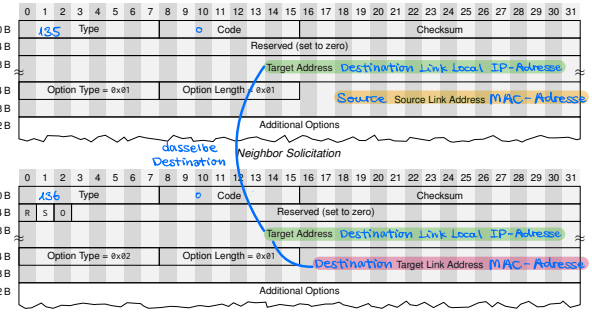
Dez	Hex	Binär	Ascii	Dez	Hex	Binär	Ascii	Dez	Hex	Binär	Ascii	Dez	Hex	Binär	Ascii
0	00	00000000	NUL	32	20	00100000	SPACE	64	40	01000000	@	96	60	01100000	`
1	01	00000001	SOH	33	21	00100001	!	65	41	01000001	A	97	61	01100001	a
2	02	00000010	STX	34	22	00100010	"	66	42	01000010	B	98	62	01100010	b
3	03	00000011	ETX	35	23	00100011	#	67	43	01000011	C	99	63	01100011	c
4	04	00000100	EOT	36	24	00100100	$	68	44	01000100	D	100	64	01100100	d
5	05	00000101	ENQ	37	25	00100101	%	69	45	01000101	E	101	65	01100101	e
6	06	00000110	ACK	38	26	00100110	&	70	46	01000110	F	102	66	01100110	f
7	07	00000111	BEL	39	27	00100111	'	71	47	01000111	G	103	67	01100111	g
8	08	00001000	BS	40	28	00101000	(	72	48	01001000	H	104	68	01101000	h
9	09	00001001	HT	41	29	00101001	)	73	49	01001001	I	105	69	01101001	i
10	0a	00001010	LF	42	2a	00101010	*	74	4a	01001010	J	106	6a	01101010	j
11	0b	00001011	VT	43	2b	00101011	+	75	4b	01001011	K	107	6b	01101011	k
12	0c	00001100	FF	44	2c	00101100	,	76	4c	01001100	L	108	6c	01101100	l
13	0d	00001101	CR	45	2d	00101101	-	77	4d	01001101	M	109	6d	01101101	m
14	0e	00001110	SO	46	2e	00101110	.	78	4e	01001110	N	110	6e	01101110	n
15	0f	00001111	SI	47	2f	00101111	/	79	4f	01001111	O	111	6f	01101111	o
16	10	00010000	DLE	48	30	00110000	0	80	50	01010000	P	112	70	01110000	p
17	11	00010001	DC1	49	31	00110001	1	81	51	01010001	Q	113	71	01110001	q
18	12	00010010	DC2	50	32	00110010	2	82	52	01010010	R	114	72	01110010	r
19	13	00010011	DC3	51	33	00110011	3	83	53	01010011	S	115	73	01110011	s
20	14	00010100	DC4	52	34	00110100	4	84	54	01010100	T	116	74	01110100	t
21	15	00010101	NAK	53	35	00110101	5	85	55	01010101	U	117	75	01110101	u
22	16	00010110	SYN	54	36	00110110	6	86	56	01010110	V	118	76	01110110	v
23	17	00010111	ETB	55	37	00110111	7	87	57	01010111	W	119	77	01110111	w
24	18	00011000	CAN	56	38	00111000	8	88	58	01011000	X	120	78	01111000	x
25	19	00011001	EM	57	39	00111001	9	89	59	01011001	Y	121	79	01111001	y
26	1a	00011010	SUB	58	3a	00111010	:	90	5a	01011010	Z	122	7a	01111010	z
27	1b	00011011	ESC	59	3b	00111011	;	91	5b	01011011	[	123	7b	01111011	{
28	1c	00011100	FS	60	3c	00111100	<	92	5c	01011100	\	124	7c	01111100	|
29	1d	00011101	GS	61	3d	00111101	=	93	5d	01011101	]	125	7d	01111101	}
30	1e	00011110	RS	62	3e	00111110	>	94	5e	01011110	^	126	7e	01111110	~
31	1f	00011111	US	63	3f	00111111	?	95	5f	01011111	_	127	7f	01111111	DEL

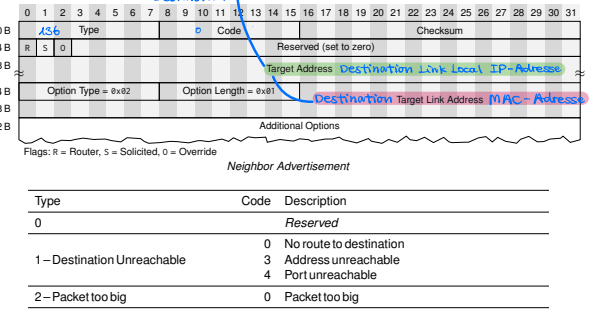
ICMPv6

Hinweis: Die folgenden ICMPv6 Nachrichten sind zu ihren ICMPv4-Pendants identisch:

- Echo Request / Reply
- Destination Unreachable
- Time Exceeded

Es gelten jedoch für ICMPv6 Types / Codes (s. unten) die Restriktionen hinsichtlich der minimalen Länge bzw. des Alignments (Padding).





Neighbor Advertisement

Type	Code	Description
0		*Reserved*
1 – Destination Unreachable	0	No route to destination
	3	Address unreachable
	4	Port unreachable
2 – Packet too big	0	Packet too big
3 – Time Exceeded	0	Hop limit exceeded in transit
	1	Fragment reassembly time exceeded
128 – Echo Request	0	Echo Request
129 – Echo Reply	0	Echo Reply
133 – Router Solicitation	0	*Neighbor Discovery Protocol (NDP)*
134 – Router Advertisement	0	*Neighbor Discovery Protocol (NDP)*
135 – Neighbor Solicitation	0	*Neighbor Discovery Protocol (NDP)*
136 – Neighbor Advertisement	0	*Neighbor Discovery Protocol (NDP)*
137 – Redirect Message	0	*Neighbor Discovery Protocol (NDP)*

Ausgewählte ICMPv6 Types/Codes

Zahlensysteme 2/2

Dez	Hex	Binär	Dez	Hex	Binär	Dez	Hex	Binär	Dez	Hex	Binär
128	80	10000000	160	a0	10100000	192	c0	11000000	224	e0	11100000
129	81	10000001	161	a1	10100001	193	c1	11000001	225	e1	11100001
130	82	10000010	162	a2	10100010	194	c2	11000010	226	e2	11100010
131	83	10000011	163	a3	10100011	195	c3	11000011	227	e3	11100011
132	84	10000100	164	a4	10100100	196	c4	11000100	228	e4	11100100
133	85	10000101	165	a5	10100101	197	c5	11000101	229	e5	11100101
134	86	10000110	166	a6	10100110	198	c6	11000110	230	e6	11100110
135	87	10000111	167	a7	10100111	199	c7	11000111	231	e7	11100111
136	88	10001000	168	a8	10101000	200	c8	11001000	232	e8	11101000
137	89	10001001	169	a9	10101001	201	c9	11001001	233	e9	11101001
138	8a	10001010	170	aa	10101010	202	ca	11001010	234	ea	11101010
139	8b	10001011	171	ab	10101011	203	cb	11001011	235	eb	11101011
140	8c	10001100	172	ac	10101100	204	cc	11001100	236	ec	11101100
141	8d	10001101	173	ad	10101101	205	cd	11001101	237	ed	11101101
142	8e	10001110	174	ae	10101110	206	ce	11001110	238	ee	11101110
143	8f	10001111	175	af	10101111	207	cf	11001111	239	ef	11101111
144	90	10010000	176	b0	10110000	208	d0	11010000	240	f0	11110000
145	91	10010001	177	b1	10110001	209	d1	11010001	241	f1	11110001
146	92	10010010	178	b2	10110010	210	d2	11010010	242	f2	11110010
147	93	10010011	179	b3	10110011	211	d3	11010011	243	f3	11110011
148	94	10010100	180	b4	10110100	212	d4	11010100	244	f4	11110100
149	95	10010101	181	b5	10110101	213	d5	11010101	245	f5	11110101
150	96	10010110	182	b6	10110110	214	d6	11010110	246	f6	11110110
151	97	10010111	183	b7	10110111	215	d7	11010111	247	f7	11110111
152	98	10011000	184	b8	10111000	216	d8	11011000	248	f8	11111000
153	99	10011001	185	b9	10111001	217	d9	11011001	249	f9	11111001
154	9a	10011010	186	ba	10111010	218	da	11011010	250	fa	11111010
155	9b	10011011	187	bb	10111011	219	db	11011011	251	fb	11111011
156	9c	10011100	188	bc	10111100	220	dc	11011100	252	fc	11111100
157	9d	10011101	189	bd	10111101	221	dd	11011101	253	fd	11111101
158	9e	10011110	190	be	10111110	222	de	11011110	254	fe	11111110
159	9f	10011111	191	bf	10111111	223	df	11011111	255	ff	11111111

Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

Formelsammlung zu Grundlagen Rechnernetze und Verteilte Systeme (GRNVS IN0010)

SS 2022, Seite 2/2, Version: 2022-05-17

α	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	$\frac{2\pi}{3}$	$\frac{3\pi}{4}$	$\frac{5\pi}{6}$	π	$\frac{3\pi}{2}$	2π
$\sin(\alpha)$	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0	-1	0
$\cos(\alpha)$	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{3}}{2}$	-1	0	1
$\tan(\alpha)$	0	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$	∞	$-\sqrt{3}$	-1	$-\frac{\sqrt{3}}{3}$	0	∞	0

$(\sin(x))' = \cos(x)$
$(\cos(x))' = -\sin(x)$

Physikalische Schicht

Physikalische Konstanten/Zusammenhänge:

Lichtgeschwindigkeit: $c_0 \approx 3 \cdot 10^8$ m/s
Relative Ausbreitungsgeschwindigkeit in Kupfer / Glas: $\nu \approx 2/3$
Relative Ausbreitungsgeschwindigkeit in Vakuum / Luft: $\nu \approx 1$
Wellenlänge im Medium: $\lambda = c/f$

Informationsgehalt und Entropie: Gedächtnislose Quelle emittiert Zeichen $x \in \mathcal{X}$, ausgedrückt durch ZV X:

Informationsgehalt von $x \in \mathcal{X}$: $I(x) = -\log_2(\Pr[X = x])$
Entropie der Quelle: $H(X) = -\sum_{x \in \mathcal{X}} \Pr[X = x] \log_2(\Pr[X = x]) = \sum \Pr[X=x] \cdot I(x)$

Fourierreihe: Kreisfrequenz $\omega = 2\pi/T$ (gleichanteil)

$$s(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\omega t) + b_k \sin(k\omega t) \text{ mit } a_k = \frac{2}{T}\int_{-T/2}^{T/2} s(t)\cos(k\omega t)\,dt \text{ (achsensymmetrisch)},\ b_k = \frac{2}{T}\int_{-T/2}^{T/2} s(t)\sin(k\omega t)\,dt \text{ (punktsymmetrisch)}.$$

Fouriertransformation: $s(t) \circ\!\!-\!\!\bullet\, S(f)$

$$S(f) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} s(t)e^{-j2\pi ft}\,dt = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} s(t)\left(\cos(2\pi ft) - j\sin(2\pi ft)\right)dt \quad (j \text{ bezeichnet die imaginäre Einheit})$$

Abtastung, Quantisierung und Rekonstruktion:

Abtasttheorem (Nyquist): $f_N = 2B$ (B ist die einseitige Grenzfrequenz im Basisband)

Abgetastetes Signal: $\hat{s}(t) = s(t)\sum_{n=-\infty}^{\infty}\delta[t - nT_a]$, mit $\delta[t - nT_a] = \begin{cases}1 & \text{für } t = nT_a\\ 0 & \text{sonst}\end{cases}$

Abtastwerte: $\hat{s}[n] = s(nT_a)$

Stufenbreite: $\Delta = \frac{b-a}{M}$, mit $M = 2^N$ Stufen bei N bit Genauigkeit

Quantisierungsstufen: $Q = \{a + \Delta/2, a + \Delta(1+1/2), \ldots, a + \Delta(M - 1 + 1/2)\}$
$\mathbb{R} \to Q,\ \hat{s}[n] \mapsto \bar{s}[n]$ (Runden)

Quantisiertes Signal: $\bar{s}(t) = \sum_{n=-\infty}^{\infty}\bar{s}[n]\cdot \text{rect}(t - nT_a)$, $\text{rect}(t) = \begin{cases}1 & \text{für } -T_a/2 \le t \le T_a/2\\ 0 & \text{sonst}\end{cases}$

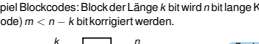
Quantisierungsfehler: $q_e(t) = s(t) - \bar{s}(t) \le \Delta/2$, wenn $a \le s(t) \le b$

Rekonstruktion: $s(t) \approx \sum_{n=-\infty}^{\infty}\bar{s}[n]\cdot \text{sinc}\left(\frac{t - nT_a}{T_a}\right)$, $\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$

Kanalbandbreite: C_{max} ist eine obere Schranke für die erzielbare Netto-Datenrate in bit/s, d. h. Übertragung redundanzfreier Daten. Dazu kann es notwendig sein, Redundanz hinzuzufügen (Kanalkodierung), was jedoch am Informationsgehalt der Nachricht nichts ändert.

(rauschfrei) Hartley: $C_H = 2B\log_2(M)$ (#Signal) (z. B. PSK m=4)
Shannon/Hartley: $C_S = B\log_2(1 + \text{SNR})$ $\quad \text{SNR} = 10\log(S) \Leftrightarrow S = 10^{\text{SNR}/10}$
Signal-to-Noise Ratio: $\text{SNR} = \frac{P_S}{P_N} = \frac{\text{Signalleistung}}{\text{Rauschleistung}} = 10\log_{10}(\text{SNR})\ \text{dB}$
Obere Schranke: $C_{max} \le \min\{C_H, C_S\}$

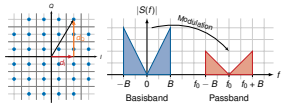
Kanalkodierung: Beispiel Blockcodes: Block der Länge k bit wird n bit lange Kanalwörter abgebildet ($n > k$). Pro Kanalwort können dafür (je nach Code) $m < n - k$ bit korrigiert werden.



Coderate: $R = k/n$

Modulation:

$$s(t) = \left(\sum_{n=0}^{\infty} d_I[n]g_T(t - nT)\right)\cos(2\pi f_0 t) - \left(\sum_{n=0}^{\infty} d_Q[n]g_T(t - nT)\right)\sin(2\pi f_0 t)$$



Sicherungsschicht und Graphen

Serialisierungszeit, Ausbreitungsverzögerung, Übertragungszeit, Bandbreitenverzögerungsprodukt:

Serialisierungszeit: $t_s = L/r$
Ausbreitungsverzögerung: $t_p = d/(\nu c_0)$ $\quad t_{s,min} = 2 \cdot t_{p,max}$
Übertragungszeit: $t_d = t_s + t_p$
Bandbreitenverzögerungsprodukt: $C = t_p r$ (Übertragungsrate)

Cyclic Redundancy Check (CRC): Addition = XOR (n bit Nullen anhängen)

Checksumme: $c(x) = m(x)x^n \bmod r(x)$, mit $n = \text{grad}\, r(x)$
Gesendete Nachricht: $s(x) = m(x)x^n + c(x)$
Überprüfung: $c'(x) = (s(x) \oplus e(x)) \bmod r(x)$, mit Fehlermuster $e(x)$ $\quad c'=0$ fehlerfrei

Adjazenz- und Distanzmatrix:

Adjazenzmatrix: $\boldsymbol{A} = (a)_{ij} = \begin{cases}1 & \exists (i,j) \in \mathcal{A}\\ 0 & \text{sonst}\end{cases}$

Distanzmatrix: $\boldsymbol{D} = (d)_{ij} = \begin{cases}c_{ij} & \exists (i,j) \in \mathcal{A}\\ 0 & \text{wenn } i = j\\ \infty & \text{sonst}\end{cases}$

min-plus-Produkt: $\boldsymbol{D}^n = \boldsymbol{D}^{n-1} \otimes \boldsymbol{D}$, mit $d_{ij}^n = \min_{k \in \mathcal{N}}\left\{d_{ik}^{n-1} + d_{kj}\right\}$, $n \ge 1$

Vermittlungsschicht

Vermittlungsarten: Übertragungszeit einer Nachricht der Länge L über n Zwischenstationen mit jeweils identischer Datenrate r über die Gesamtdistanz d:

Leitungsvermittlung: $T_{LV} = t_s + 4t_p = \frac{L}{r} + \frac{4d}{\nu c_0}$

Nachrichtenvermittlung: $T_{NV} = (n+1)t_s + t_p = (n+1)\frac{L_H + L}{r} + \frac{d}{\nu c_0}$, L_H = Länge des Nachrichtenheaders

Paketvermittlung: $T_{PV} = \frac{1}{r}\left(\left\lceil\frac{L}{\rho_{max}}\right\rceil L_h + L + n(L_h + \rho_{max})\right) + \frac{d}{\nu c_0}$, L_h = Länge der Paketheader

Round Trip Time (RTT): RTT zwischen den Knoten $s, t \in \mathcal{N}$ über den Pfad $\mathcal{P} = \{(s,1),(1,2),\ldots,(n,t)\}$ und den i. A. nicht symmetrischen Rückweg $\mathcal{P}'$:

RTT (allgemein): $\text{RTT}(s,t) = \sum_{(i,j)\in\mathcal{P}}(t_s(i,j) + t_p(i,j)) + \sum_{(i,j)\in\mathcal{P}'}(t_s(i,j) + t_p(i,j))$

RTT (symmetrische Pfade): $\text{RTT}(s,t) = 2\sum_{(i,j)\in\mathcal{P}}(t_s(i,j) + t_p(i,j))$

Spezielle IP-Adressen / -Adressbereiche:

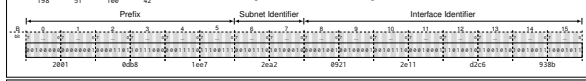
Adressbereich	Funktion
0.0.0.0/8	Hosts in diesem Netzwerk
127.0.0.0/8	Loopback, speziell 127.0.0.1
10.0.0.0/8	private Adressen
172.16.0.0/12	private Adressen
192.168.0.0/16	private Adressen
169.254.0.0/16	Automatic Private IP Addressing
255.255.255.255/32	Global Broadcast

Adressbereich	Funktion
::/128	nicht-spezifizierte Adresse
::1/128	Loopback
fe80::/10	Link-Local Adressen
fc00::/7	Unique-Local Unicast Adressen
ff00::/8	Multicast Adressen
ff02::1/128	All Nodes
ff02::1:ff00:0/104	Solicited Node Adressen

IPv4/6 Adressformat: (Beispiele)



Bei IPv4 unterscheidet man nicht zwischen Präfix und Subnetz (das Präfix definiert das jeweilige Subnetz). Bei IPv6 spricht man zusätzlich von einem *Subnet Identifier*, der zusammen mit dem Präfix das jeweilige Subentz identifiziert. Die Schreibweise <address>/N gibt dabei immer die Länge des Netzanteils an.



Transportschicht

Schiebefensterprotokolle

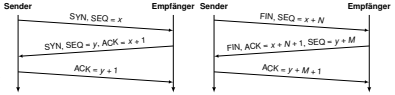
Kardinalität Sequenznummernraum: N. Maximale Größe des Sendefensters w_s um Verwechselungen zu vermeiden:

Go-Back-N: $w_s \le N - 1$
Selective Repeat: $w_s \le \left\lfloor\frac{N}{2}\right\rfloor$

Fenster bei TCP

Empfangsfenster: w_r
Staukontrollfenster: w_c
Sendefenster: $w_s = \min\{w_r, w_c\}$

TCP-Handshake und TCP-Teardown



TCP Durchsatz in der Congestion Avoidance Phase. Annahme: Segmentverlust im Netzwerk ab $w_s \ge x \cdot \text{MSS}$.

Zeit zwischen Segmentverlust: $T = \left(\frac{x}{2} + 1\right) \cdot \text{RTT}$
Anzahl gesendeter Segemente in T: $n = \frac{3}{8}x^2 + \frac{3}{4}x$
Verlustrate: $\theta = \frac{1}{n}$
Durchsatz: $r_{TCP} = \frac{n \cdot \text{MSS}}{T} \cdot (1 - \theta)$

Anwendungsschicht

Präfixfreie Codes

Gültige Codewörter eines *präfixfreien Code* sind niemals Präfix eines anderen Codeworts desselben Codes.
Ein *optimaler* präfixfreier Code minimiert die mittlere Codewortlänge

$$\sum_{i \in \mathcal{A}} p(i) \cdot |c(i)|,$$

wobei $p(i)$ die Auftrittswahrscheinlichkeit von $i \in \mathcal{A}$ und $c(i)$ die Abbildung auf ein entsprechendes Codewort bezeichnen.

DNS Resource Records

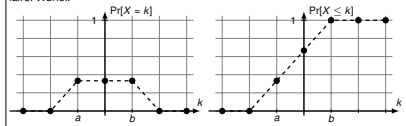
Record-Typ	Funktion
SOA	(Start of Authority) markiert die Wurzel einer Zone
NS	geben die FQDNs der für die Zone autoritativen Nameserver an
A	assoziieren einen FQDN mit einer IPv4-Adresse
AAAA	assoziieren einen FQDN mit einer IPv6-Adresse
CNAME	Alias, verweist auf ein „Canonical Name", welcher wiederum ein FQDN ist
MX	geben den Mailserver als FQDN einer Domain an
TXT	assoziieren einen FQDN mit einem String (Text)
PTR	assoziieren eine IPv4- oder IPv6-Adresse mit einem FQDN (Reverse DNS)

Reverse DNS Zonen

IPv4: `in-addr.arpa.`, IPv6: `ip6.arpa.`

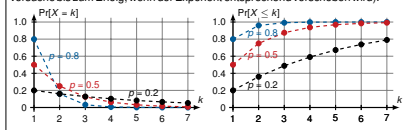
Wahrscheinlichkeitsverteilungen

Diskrete Gleichverteilung: $X \sim \text{U}(a,b)$:
Drückt die Wahrscheinlichkeit für das Auftreten eines bestimmten von mehreren gleichwahrscheinlichen Ereignissen aus, z. B. fairer Würfel.



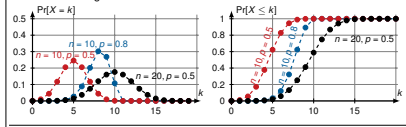
$\Pr[X = k] = \frac{1}{b - a + 1}$
$\Pr[X \le k] = \frac{k - a + 1}{b - a + 1}$
$\text{E}[X] = \frac{a + b}{2}$
$\text{Var}[X] = \frac{(b - a + 1)^2 - 1}{12}$

Geometrische Verteilung: $X \sim \text{Geo}(p)$:
Drückt ein zeitdiskretes Warteproblem aus, z. B. zählt die Anzahl der Versuche bis zum Erfolg (bzw. die Anzahl erfolgloser Versuche bis zum Erfolg, wenn der Exponent entsprechend verschoben wird).



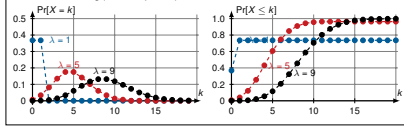
$\Pr[X = k] = (1-p)^{k-1}p$
$\Pr[X \le k] = 1 - (1-p)^k$
$\text{E}[X] = \frac{1}{p}$
$\text{Var}[X] = \frac{1-p}{p^2}$

Binomialverteilung: $X \sim \text{Bin}(n,p)$:
Drückt die Wahrscheinlichkeit für $0 \le k \le n$ Erfolge bei konstanter Erfolgswahrscheinlichkeit p aus, z. B. Lotto. Für $n \to \infty$ und $p \to 0$ erhählt man die Poissonverteilung. Für $n \ge 10$ und $p < 0.5$ kann man die Poissonverteilung als Näherung für die Binomialverteilung verwenden.



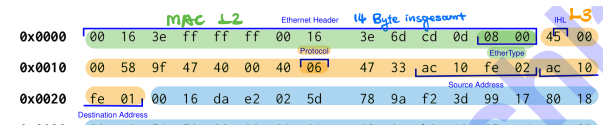
$\Pr[X = k] = \binom{n}{k}p^k(1-p)^{n-k}$
$\Pr[X \le k]$ *keine geschlossene Form*
$\text{E}[X] = np$
$\text{Var}[X] = np(1-p)$

Poissonverteilung: $X \sim \text{Po}(\lambda)$:
Zählt das Auftreten unabhängiger und gleich verteilter Ereignisse mit Rate λ. Stellt für $\lambda = np$ den Grenzwert der Binomialverteilung ($n \to \infty, p \to 0$) dar.



$\Pr[X = k] = \frac{\lambda^k}{k!}e^{-\lambda}$
$\Pr[X \le k]$ *keine geschlossene Form*
$\text{E}[X] = \lambda$
$\text{Var}[X] = \lambda$

L2 MAC
L3 IPV4/IPV6
L4 TCP/UDP



L3-PDU Ethertype in Ethernet Header
L3-SDU = L4-PDU = Protokoll/NextHeader in IPV4/IPV6 Header

k) Berechnen Sie die Länge der L7-PDU in Byte.

payload length Feld im IP Header = 0x01a3 → 419 B L3-SDU
TCP Header Länge (inklusive Options) → 32 B
→ L7-PDU = 419 B - 32 B = 387 B

-> kompakte Schreibweise
- führende Nullen werden weg gelassen: 2001:db8:1:d:0:f0:0:0 /64
- der größte konsekutive Block von mindesten 2 „Nuller"-Blöcken kann durch :: abgekürzt werden: 2001:db8:1:d:0:f0::
- Beispiel Loopback-Adresse: **::1/128**

Addresse Länge
IPV6 128 bits
IPV4 32 bits

$2^{128-64} = 2^{64}$ Adresse

- Abtastung ist die Diskretisierung eines kontinuierlichen Signals im Zeitbereich ohne Runden.
- Quantisierung ist die Diskretisierung eines Signals in Signalstufen, d. h. im Wertebereich mit Runden.

Schritt i Router A Eintrag D bestimmen ←
→ Schritt i-1 Router B Eintrag D + $Distanz_{AB}$ im Bild
→ Schritt i-1 Router C Eintrag D + $Distanz_{AC}$ im Bild
} vergleichen, kleinere eintragen

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

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

Stateless Address Autoconfiguration (SLAAC):
-> Host generiert sich **link-local IPv6-Adresse** (für Interface) aus MAC-Adresse
-> Das **Präfix ist fe80::/10**
-> Der **Subnet Identifier** (die folgenden 54 bit) werden **auf 0 gesetzt**
-> Die verbleibenden 64 bit stellen den Interface Identifier dar, welcher aus der MAC-Adresse des jeweiligen Interfaces als modifizierter EUI-64 Identifier generiert wird:
- Die ersten **24 bit** sind der **OUI** der MAC-Adresse (**vorletztes Bit des 1. Oktetts** des OUI invertiert) — erste Byte
- Die nachfolgenden **16 bit** werden mit **ff:fe** „gestopft"
- Die restlichen **24 bit** werden mit dem **Device Identifier** der MAC-Adresse aufgefüllt

- PC1: `af:fe:14:af:fe:20` → `fe80::adfe:14ff:feaf:fe20`

af 1010 1111
ad 1010 1101

b) Bestimmen Sie die Global-Unique Adressen von PC1 und PC2. Nehmen Sie dazu an, dass Router R1 mit dem Präfix `2001:db8:1::/64` und Router R2 mit `2001:db8:2::/64` konfiguriert sind.

Die Herleitung geschieht analog zu den Link-Local-Adressen, allerdings mit dem Präfix des jeweiligen Routers, welche über Router Advertisements PC1 und PC2 bekannt gemacht werden.

- `af:fe:14:af:fe:20` → `2001:db8:1:0:adfe:14ff:feaf:fe20`
- `af:fe:14:af:fe:25` → `2001:db8:2:0:adfe:14ff:feaf:fe25`

af 1010 1111
ad 1010 1101

***Mapping** von Multicast IPv6 Adressen auf MAC-Adressen* (6 Bytes = 48 bits)
1. Die ersten 2 Oktette der MAC-Adresse werden auf 33:33 gesetzt
   - **letztes** Bit des ersten Oktetts ist gesetzt → **Multicast** — Bedingung für 33:33
   - **vorletztes** Bit des ersten Oktetts ist gesetzt → **locally administered (bei IPv6 andersrum)**
2. Die letzten 4 Oktette der Ethernetadresse werden die letzten 4 Oktette der IPv6 Multicastadresse

h) Bestimmen Sie die Destination-MAC-Adresse des ersten übertragenen Rahmens.

Der Rahmen enthält ein IPv6 Neighbor Solicitation Paket. Diese werden laut Vorlesung an die entsprechende Solicited-Node Adress geschickt. Da diese Multicast Adressen sind, berechnet sich die Destination-MAC-Adresse aus der Solicited-Node Adresse.
Es soll die MAC Adresse für R1 mit der IP `fe80::adfe:14ff:feaf:fe21` erfragt werden. Mit dem Solicited-Node Prefix `ff02::1:ff00:0/104` und den letzen 24 bit der Ziel Adresse ergibt sich die Solicited-Node Adresse `ff02::1:ffaf:fe21`. Diese liegt im Prefix `ff00::/8` und ist somit eine Multicast Adresse. Deshalb wird die zugehörige MAC aus dem Prefix `33:33` und den letzen 32 bit der Ziel Adresse `ff02::1:ffaf:fe21` berechnet und es ergibt sich `33:33:ff:af:fe:21`.

Neighbour Solicitation & Neighbour Advertisement Größe = 32 Byte

c)* An welcher Stelle im Netzwerk wird die Fragmentierung stattfinden?

Direkt an PC1, da bei IPv6 keine Fragmentierung an Routern stattfindet.

f)* Begründen Sie, an welcher Stelle im Netzwerk werden die Fragmente reassembliert werden.

Erst der Empfänger, hier also PC2, reassembliert die Fragmente wieder. Tatsächlich kann i. A. kein anderer Knoten die Reassemblierung durchführen, da die Fragmente jeweils einzelne und voneinander unabhängige Pakete darstellen. Dies bedeutet insbesondere, dass sie unabhängig voneinander geroutet werden und daher u. U. verschiedene Wege zum Ziel nehmen können – das sieht man aus dem einfachen Beispiel in Abbildung 2.1 natürlich nicht, da es hier nur einen Pfad zwischen PC1 und PC2 gibt.

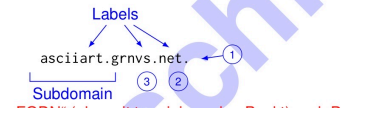
- **Go-Back-N**: Empfänger akzeptiert nur das nächste erwartete Segment. Segmente, die out-of-order angkommen werden ignoriert.
- **Selective Repeat**: Alle Segmente werden zwischengepuffert. Nur unbedingt notwendige Segmente werden gesendet.

a)* Was ist der Unterschied zwischen einem vollqualifizierten Domain Name (FQDN) und einem nicht (voll)qualifizierten?

Ein FQDN endet stets mit ., d. h. der Wurzel des Name Spaces. Ein nicht-qualifizierter Domain Name hingegen kann ein einzelnes Label oder eine geordnete Liste durch Punkte getrennter Labels sein, die relativ zu einer anderen Wurzel als . zu sehen sind.

b)* Benennen Sie die einzelnen Bestandteile des FQDNs, sofern es dafür gängige Bezeichnungen gibt.

1. Root (Beginn des Namensraums)
2. Top Level Domain (TLD)
3. Second Level Domain



c)* Erläutern Sie den Unterschied zwischen einem *Resolver* und einem *Nameserver*.

Nameserver sind autoritativ für eine oder mehrere Zonen („Bereiche"), d. h. sie besitzen eine gültige und aktuelle Kopie der gesamten Zone, für die sie autoritativ sind.
Resolver hingegen extrahieren mittels eine Reihe iterativer Anfragen an die jeweils autoritativen Nameserver die benötigten Information aus dem DNS und geben diese an den anfragenden Client zurück. Resolver können Einträge für begrenzte Zeit cachen, so dass bei erneuter Anfrage derselben Resource Records der Prozess nicht wiederholt werden muss.

d)* Welche Funktion erfüllen `d.root-servers.net` und `a.gtld-servers.net`?

Der Root-Nameserver ist autoritativ für die Rootzone, d. h. er kennt die Nameserver, welche für die einzelnen TLDs verantwortlich sind, so z. B. `a.gtld-servers.net` als einen der autoritativen Nameserver für net-Domains.
`a.gtld-servers.net` kennt wiederum die zuständigen Nameserver für alle Second-Level-Domains unterhalb der net-TLD.

e)* Erklären Sie den Unterschied zwischen iterativer und rekursiver Namensauflösung.

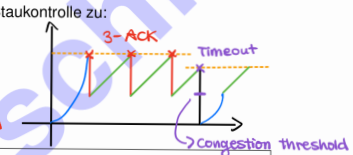
Rekursive Namensauflösung bedeutet, dass eine DNS-Anfrage an einen Resolver gestellt wird. Dieser wird das endgültige Ergebnis zurücksenden.
Bei iterativer Auflösung hingegen werden schrittweise die autoritativen Nameserver der einzelnen Zonen angefragt.

a)* Diskutieren Sie die Unterschiede zwischen Fluss- und Staukontrolle. Welche Ziele werden mit dem jeweiligen Mechanismus verfolgt?

- **Flusskontrolle:**
  Verhinderung von Überlastsituation beim Empfänger
- **Staukontrolle:**
  Reaktion von Überlastsituation im Netz

b) Ordnen Sie die folgenden Begriffe jeweils der TCP-Fluss- bzw. Staukontrolle zu:

- Slow-Start — exponentiale Phase
- Empfangsfenster
- Congestion-Avoidance — lineare Phase
- Multiplicative-Decrease — bei Segmentverlust halbieren

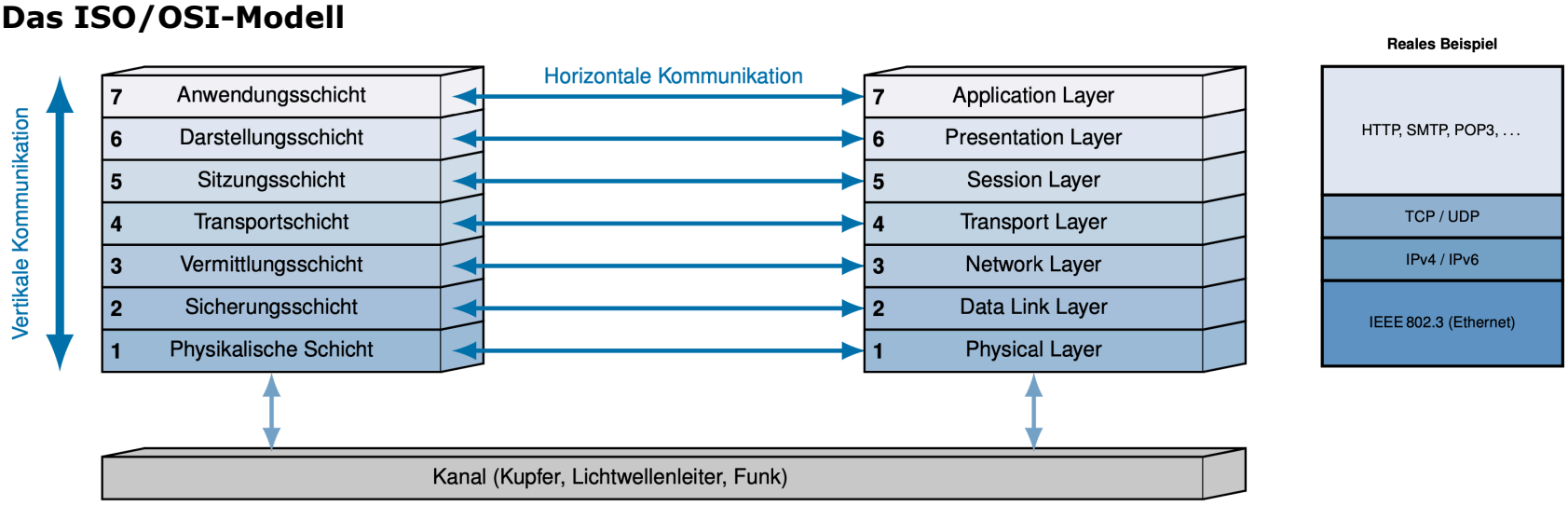


Zur Flusskontrolle gehört lediglich das Empfangsfenster, da über dieses der Empfänger dem Sender mitteilen kann, wie viel Daten er maximal auf einmal senden darf.
Die übrigen Begriffe gehören alle zur Staukontrolle, wobei Slow-Start und Congestion-Avoidance die beiden Staukontrollphasen einer TCP-Verbindung sind. Als Multiplicative-Decrease hingegen bezeichnet man das Halbieren des Staukontrollfensters bei Verlust eines Segments.

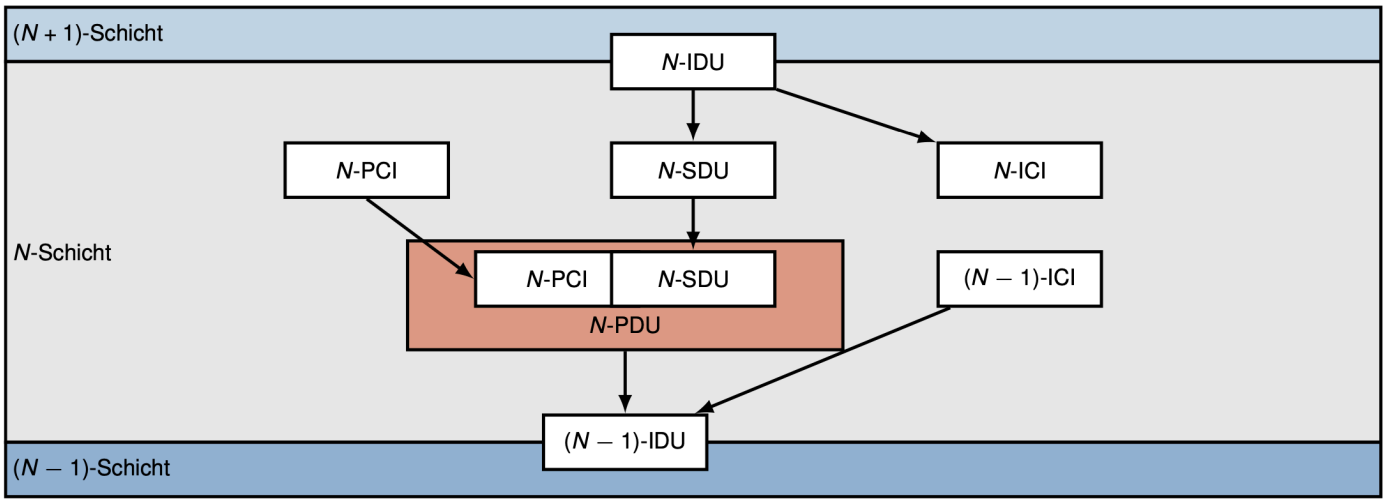
GRNVS-Lernzettel

Kapital 0: Einführung

Das ISO/OSI-Modell



Abstraktion der Kommunikation zwischen Schichten



Kapital 1: Physikalische Schicht

Informationsgehalt I des Zeichens x I(x) = −log2 p(x) wobei p(x) = Pr[X = x]

Entropie H = mittlere Informationsgehalt einer Quelle $H(X) = \sum_{x \in \mathcal{X}} p(x) I(x) = -\sum_{x \in \mathcal{X}} p(x) \log_2 (p(x))$

Bedingte Entropie - Unsicherheit über Y, wenn X bekannt $H(Y|X) = \sum_{x \in \mathcal{X}} p(x) H(Y|X = x) = -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log_2 p(y|x)$

Verbunddichte p(x,y) = Pr[X = x|Y = y] * Pr[Y = y] = Pr[X = x|Y = y] * Pr[X = x]

Verbundentropie $H(X,Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log_2 p(x,y)$ $H(X,Y) = H(X) + H(Y|X)$

bedingte Entropie $H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{XY}(X = x, Y = y) \log_2 (p_Y(Y = y|X = x))$

1. Deterministische, diskrete Quelle, welche stets das Zeichen 'A' emittiert:

   Q —X→ AAAAA...

   $I(\text{A}) = -\log_2 (\Pr[X = \text{A}]) = -\log_2(1) = 0 \text{ bit}$

2. Binäre Quelle, welche auf nicht vorhersehbare Weise die Zeichen '0' oder '1' emittiert:

   Q —X→ 01101...

   $I(0) = -\log_2 (\Pr[X = 0]) = -\log_2(0{,}5) = 1 \text{ bit}$

   $I(1) = -\log_2 (\Pr[X = 1]) = -\log_2(0{,}5) = 1 \text{ bit}$

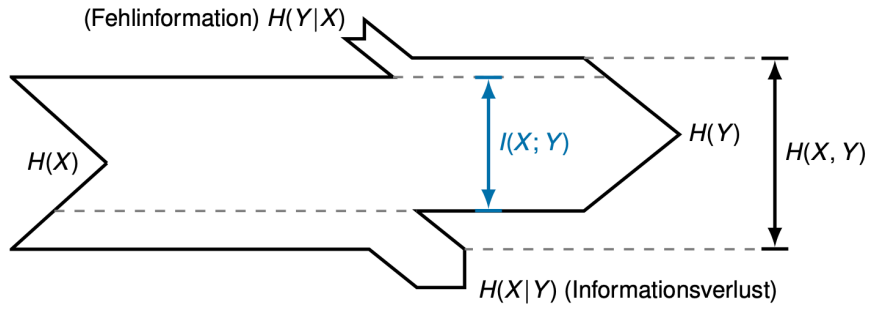
   Die Entropie $H(X) = \sum_i p_i I(x_i)$ dieser Quelle beträgt

   $H(X) = -(p_0 \log_2(p_0) + p_1 \log_2(p_1)) = -(-0{,}5 - 0{,}5) = 1 \text{ bit/Zeichen.}$

Informationstheoretisches Modell eines gedächtnislosen Kanals

Empfangsentropie $H(Y) = H(X) - H(X|Y) + H(Y|X)$

Transinformation $I(X; Y) = H(X) - H(X|Y)$
$= H(Y) - H(Y|X)$



Signaldarstellung

- **periodische** Signale -> Darstellung mittels **Fourierreihe** (**diskrete** Spektrum) $s(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos(k\omega t) + b_k \sin(k\omega t))$
  - k-te Summenglied = k-te Harmonische; **Kreisfrequenz ω** = 2π/T wobei π ist pi
  - **Gleichanteil $a_0/2$** = **Vertikale Verschiebung** des gesamten Signals
  - s(t) in Phase mit einer Sinus-/ Kosinusschwinung -> Phase = **horizontale Verschiebung** einer Schwingung
  - **Koeffizienten (Gewichte)** $a_k = \frac{2}{T}\int_0^T s(t)\cos(k\omega t)\,dt$ und $b_k = \frac{2}{T}\int_0^T s(t)\sin(k\omega t)\,dt$ achsensymmetrisch punktsymmetrisch punktsymmetrisch zu (T/2, 0)

Punktsymmetrisch Signale bestehen nur aus punktsymmetrisch Basisfunktion (仅Sinusanteile) ⇒ $a_k = 0$ für alle k

Achsensymmetrisch Signale bestehen nur aus achsensymmetrisch Basisfunktion (仅Kosinusanteile) ⇒ $b_k = 0$ für alle k

- **nicht-periodische Signale** -> Darstellung mittels **Fouriertransformation** (**Kontinuierliches** Spektrum)
  - Fourier-Transformierte von s(t) $s(t) \circ\!\!-\!\!\bullet\, S(f) = \frac{1}{\sqrt{2\pi}}\int_{t=-\infty}^{\infty} s(t)e^{-j2\pi ft}\,dt = \frac{1}{\sqrt{2\pi}}\int_{t=-\infty}^{\infty} s(t)(\cos(2\pi ft) - j\sin(2\pi ft))\,dt$

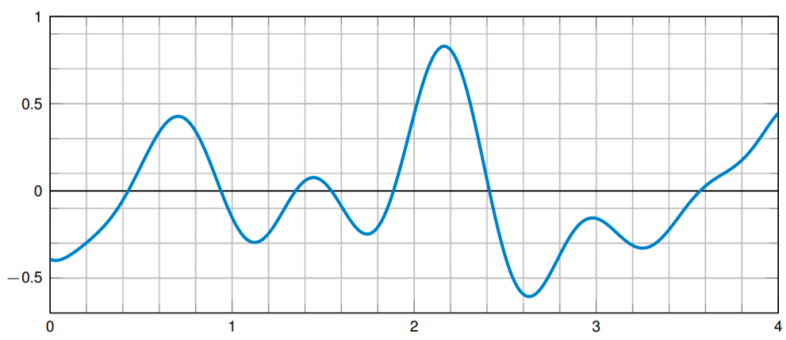
Abtastung = Diskretisierung von Signalen im **Zeitbereich**

- Mittels der Abtastwerte s^[n] möglich, ursprüngliche Signal s(t) zu **rekonstruieren**: $s(t) \approx \sum_{n=-\infty}^{\infty} \hat{s}[n] \cdot \mathrm{sinc}\left(\frac{t - nT_a}{T_a}\right)$
- Abtastwerte sind **Stützstellen** und dienen als Gewichte für eine passende Ansatzfunktıon
- Summe der gewichteten Ansatzfunktionen nähert sich dem ursprünglichen Signal (abhängig von #Summenglied)
- Abtastung des Signals s(t) in Abständen T_a entspricht der periodischen Wiederholung seines Spektrums S(f) in Abständen von $f_a = 1/T_a$

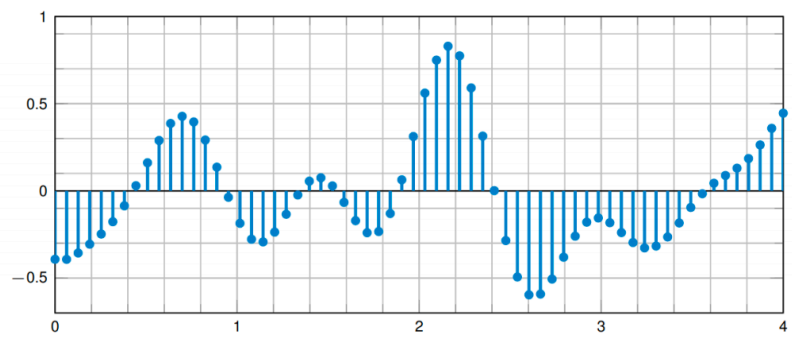
- **Abtasttheroem** von Shannon und Nyquist: Ein auf $|f| \leq B$ begrenztes Signal muss **mindestens** mit der Frequenz 2B (Nyquist-Rate) abgetastet werden, um das Signal verlustfrei rekonstruieren zu können (damit keine Information verlohren geht); **maximal** 2B unterscheidbare Symbolen pro Zeiteinheit (die über Kanal sicher übertragen wird)
  - **Aliasing:** falls $f_a < 2B$, überlappen sich die periodischen Wiederholungen des Spektrums)

Quantisierung = Diskretisierung von Signalen im **Wertbereich** in **M Signalstufen**

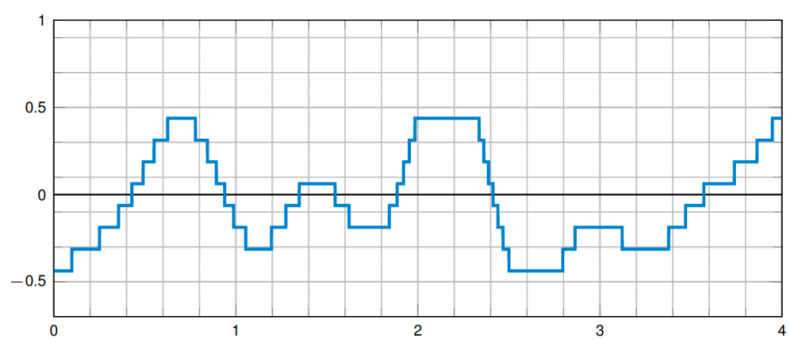
- Jede Signalstufe erhält ein zugehöriges Codewort der Breite N (**$M = 2^N$ Signalstufe**) -> Runden auf nächste Stufe
- **Stufenbreite** Δ = b-a/M; **Quantisierungsfehler** $q_{max} = \Delta/2$ (**innerhalb** I_Q); unbeschränkt (**außerhalb** I_Q)
- **Gray-Code:** Benachbarte Codewörter unterscheiden sich jeweils nur in einer binären Ziffer (Hamming-Distanz = 1)



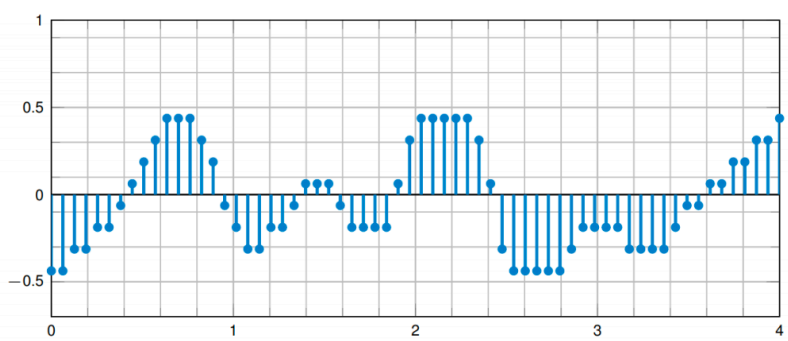
normalerweise (a) Analog $s(t)$ zeitkontinuierlich & wertkontinuierlich



(b) Zeitdiskret und wertkontinuierlich $\hat{s}[n]$



(c) Zeitkontinuierlich und wertdiskret $\bar{s}(t)$



(d) Digital $s[n]$ zeitdiskret & wertdiskret

Übertragungskanal

- Beim Übertragen eines Signals über einen Kanal passieren folgende Effekte
  - **Dämpfung:** Die Amplitude des Signals wird **verkleinert** - am Ausgang kleiner als am Eingang
  - **Tiefpassfilterung:** Signalflanken werden **abgedämpft** - höhere Frequenzen stärker gedämpft als niedrige
  - **Verzögerung:** Das Signal kommt **zeitlich verzögert** an
  - **Rauschen:** Es kommen **Störsignale** hinzu - in Form von Additive White Gaussian Noise (AWGN)
    - **SNR** (Signal to Noise Ratio) = Signalleistung P_S/ Rauschleistung P_N (oft in **$10 \cdot \log_{10}(SNR)$** dB angegeben)
- Ein Filter lässt nur bestimmte Teile eines Frequenzbandes „passieren" (|f| < B passieren, sonst gesperrt)
  - Ein **Tiefpassfilter** lässt nur tiefe (lange Wellenlänge, kleines f [Hz]) Frequenzen eines Signals durch
  - Ein **Bandpassfilter** lässt nur ein bestimmtes Intervall von Frequenzen durch - Beispiel: Radioempfänger
  - Ein **Hochpassfilter** lässt nur ein hohe (kurze Wellenlänge, großes f [Hz]) Frequenzen durch

- **Hartley Gesetz: maximale Übertragungsrate** über einen Kanal der Bandbreite B mit M unterscheidbaren Signalstufen (**Kanalkapazität** nach oben beschränkt): **$C_H = 2B \log_2(M)$** bit
  - B groß -> besserer Kanal, weil das Signal durch den Tiefpasseffekt weniger verzerrt wird
  - B klein -> schlechterer Kanal, weil das Signal durch den Tiefpasseffekt stärker verzerrt wird

- **Shannon-Hartley-Theorem:** Auf einem Kanal der Bandbreite B mit additiven weißen Rauschen mit P_S und P_N beträgt die obere Schranke **$C_S = B \log_2(1+ P_S/P_N)$** bit für die erreichbare Datenrate

-> **C_S** berücksichtigt nur additives Rauschen des Kanals, aber keine Quantisierungsfehler

-> **C_H** berücksichtigt nur die Signalstufen und damit das „Quantisierungsrauschen", aber keine Kanaleinflüsse

Kanalkapazität C durch zwei Faktoren beschränkt: Anzahl der unterscheidbaren Symbole M, SNR

obere Schranke für die **tatsächliche Kanalkapazität**: **$C < \min\{C_H, C_S\}$** = $\min\{2B \log_2(M), B \log_2(1 + SNR)\}$ bit

Nachrichtenübertragung

- **Quellenkodierung:** durch Abbildung von Bitsequenzen auf Codewörter **Redundanz entfernen (verlustlose Datenkompression)**

- **Kanalkodierung (Fehlererkennung):** gezielt Redundanz hinzufügen, damit Übertragungsfehler detektiert und korrigiert werden können
  - **Blockcodes** der Länge k werden **in Kanalwörter** der Länge n > k **übersetzt** (n-k bit für Fehlererkennung und Rekonstruktion); **Coderate** R = k/n
  - **Bsp.** Repetition Code (0->000, 1->111) -> **Problem:** Die zu sendende Bit-Anzahl wird größer -> erzielbare Datenrate sinkt

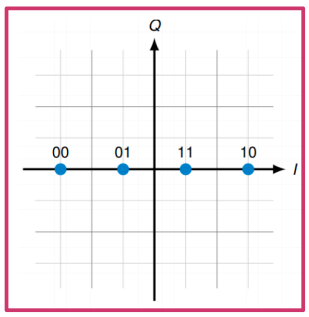
- **Leitungskodierung:** Leitungscode = definiert die Abfolge von Grundimpulsen (Sendeimpuls) , welche Bits oder Gruppen von Bits repräsentieren
  - **Eigenschaften: wichtige** -> Anzahl der Signalstufen, Anzahl kodierter bits pro Symbol, Schrittgeschwindigkeit (Einheit bd); **optional** -> Taktrückgewinnung, Gleichstromfreiheit, Steuerzeichen
  - **Non-Return-To-Zero (NRZ)** -> keine Taktrückgewinnung, keine Gleichstromfreiheit, relativ breites Spektrum
  - **Return-To-Zero (RZ)** -> Taktrückgewinnung (durch erzwungenen Pegelwechsel), aber keine Gleichstromfreiheit, weniger effizient (2 Symbole/bit), Breiteres Spektrum als NRZ
  - **Manchester-Code** -> Taktrückgewinnung & Gleichstromfreiheit (jeder Grundimpuls für sich schon gleichstromfrei), sehr breites und langsam abklingendes Spektrum
  - **Multi-Level-Transmit 3 (MLT3)** -> keine Taktrückgewinnung, keine Gleichstromfreiheit, schmales Spektrum

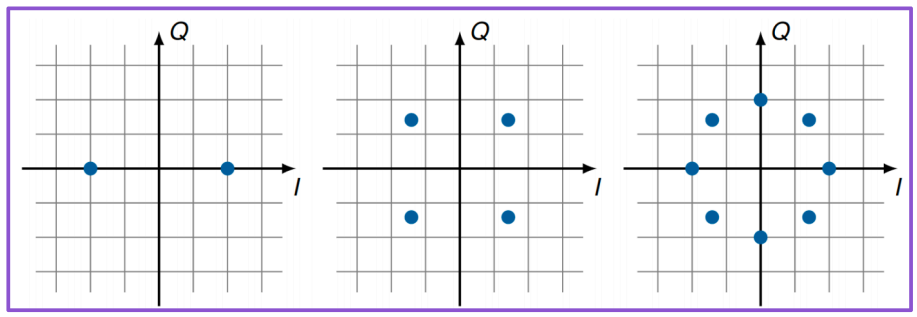
—> Empfänger erkennt den Beginn/ Ende einer Nachricht durch **Coderegelverletzung (idle)** oder **Steuerzeichen**

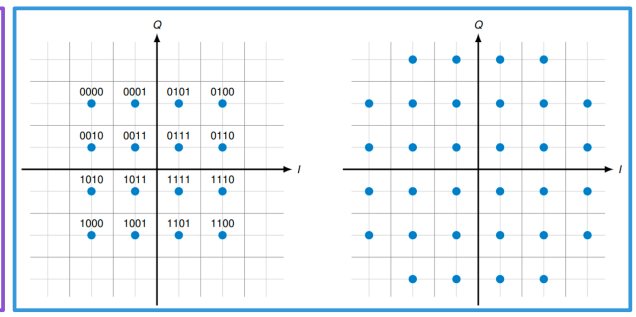
- **Modulation**
  - **Frequency Division Multiplex (FDM):** mehrere Übertragungen teilen sich zeitgleich ein Kanal (higher bandwith requirements) - zu übertragenes Signal wird mit sogenannter Trägerfrequenz multipliziert
  - Trägerfrequenz in gewichtete **Sinus**anteile (**Quadratur**-Anteil) und **Kosinus**anteile (**Inphase**-Anteil) zerlegen

y-Achse x-Achse

cos • **ASK (Amplitude Shift Keying)** (bsp. 4-ASK = 4 verschiedene Symbole der Breite N=2) - Problem: noise
  - Alle Symbole liegen im IQ-Diagramm auf der Inphase-Achse, nur Amplitude der Trägerfrequenz moduliert
• **PSK (Phase-Shift-Keying) -** bsp. 2-PSK, 4-PSK, 8-PSK
  - Alle Symbole liegen im IQ-Diagramm auf dem Einheitskreis, Nur Phase der Trägerfrequenz moduliert
  - Problem: mehr Signalstufen -> schwieriger zu unterscheiden, höhere Fehlerrate
cos+sin • **QAM (Quadratur-Amplituden-Modulation)** - bsp. 16-QAM, 32-QAM
  - Alle Symbole sind in einem Gitter angeordnet, Phase & Amplitude der Trägerfrequenz werden modifiziert
  - Datenrate (Bandbreite) lässt sich auf diese Weise verdoppeln, aber reduziert Fehlerresistenz







Übertragungsmedien
- **leitungsgebundene** Übertragung vs. **nicht-leitungsgebundener** Übertragung
- **akustische** vs. (überwiegend) **elektromagnetische** Wellen

Elektromagnetische Wellen
- Ausbreitung im Vakuum mit Lichtgeschwindigkeit $c_0 \approx 3 * 10^8$ m/s
- Im Gegensatz zu Schallwellen wird kein Medium zur Ausbreitung benötigt
- Das verwendete Medium hat Einfluss auf die Ausbreitungsgeschwindigkeit $c = v*c_0$
  - wobei $v \approx 0.7$ in Lichtwellenleitern, $v \approx 2/3$ in Koaxialleitern
- Frequenz f ergibt sich aus Lichtgeschwindigkeit und Wellenlänge im Medium zu $f = c/\lambda = c_0/\lambda_0$
- Beim Übergang vom Vakuum in ein Medium bleibt die Frequenz f konstant, Wellenlänge und Ausbreitungsgeschwindigkeit ändern sich proportional zueinander

Koaxialleiter
- Eingesetzt u. a. für IEEE 802.3a („10Base2 Ethernet“, 10 Mbit/s)
- Einsatzbereiche: TV-Kabelnetz, Hochfrequenztechnik\

Twisted-Pair-Kabel
- 2 oder 4 Aderpaare aus Kupferlitzen
- Zweite Ader eines Paares führt inversen Signalpegel (differentielle Kodierung)
- Verwendung: Lokale Netzwerke (Ethernet Standard), Telefonanschlüsse
- Schirmung hat Einfluss auf Signalqualität, Flexibilität der Kabel

Optische Leiter
- Licht wird innerhalb des Faserkerns weitergeleitet
- **Single-Mode-Fasern** vermeiden Streuung durch sehr geringen Kerndurchmesser
  - geringe Verluste, aber sehr empfindlich (Kabelbruch)
- **Multi-Mode-Fasern** haben einen größeren Kerndurchmesser und neigen daher zum Streuen
  - höhere Verluste, aber weniger empfindlich
- Vorteile gegenüber elektrischen Leitern: hohe Datenraten möglich, weite Strecken überbrückbar, kein Übersprechen

Kapitel 2: Sicherungsschicht

Direktverbindungsnetz
- alle angeschlossenen Knoten sind **direkt erreichbar**
- werden mittels **einfacher Adressen** in Schicht 2 identifiziert
- es findet **keine Vermittlung** statt, aber **einfache Weiterleitung** möglich **(Bridging oder Switching)**, keine Router!

Aufgaben der Sicherungsschicht:

- **Steuerung des Medienzugriffs:**
  - Hubs (Intern werden alle angeschlossenen Computer zu einem Bus verbunden)
  - Gleichzeitiges Senden von zwei Stationen führt zu **Kollisionen** und daher zum **Verlust** von Nachrichten
- **Prüfung übertragener Nachrichten auf Fehler:**
  - Trotz Kanalkodierung treten **Übertragungsfehler** auf (müssen erkannt werden)
  - Defekte Nachrichten werden nicht an höhere Schichten weitergegeben
  - Wiederholung einer Übertragung ist häufig Aufgabe höherer Schichten
  - Die meisten Fehler heutzutage werden durch **TCP** behoben
- **Adressierung innerhalb von Direktverbindungsnetzen:**
  - Der jeweilige Empfänger einer Nachricht muss entscheiden können, ob die Nachricht für ihn bestimmt ist

Darstellung von Netzwerken als Graphen

-> Darstellung von Netztopologien

- **gerichtete** Graphen $G = (N,A)$
  - **asymmetrisches** Netzwerk
  - N eine Menge von Knoten (Nodes)
  - $A = \{(i,j) \mid i,j \in N \wedge i, j$ sind gerichtet verbunden$\}$ eine Menge gerichteter Kanten (Arcs)
- **ungerichtete** Graphen $G = (N,E)$
  - **symmetrisches** Netzwerk
  - N eine Menge von Knoten
  - $E = \{\{i,j\} \mid i,j \in N \wedge i, j$ sind ungerichtet verbunden$\}$ Menge ungerichteter Kanten (Edges)

-> Ungerichtete Graphen können als gerichtete Graphen mit sym. Kanten verstanden werden

Netztopologie

- beschreibt die **Struktur**, wie Knoten miteinander verbunden sind
  - physikalische Topologie
  - logische Topologie
- Bsp.: Punkt zu Punkt, Kette, Stern, Vermaschung, Baum (meist logisch), Bus

Adjazenz- und Distanzmatrix

- Adjazentmatrix: gibt an, ob Knoten i mit Knoten j verbunden ist
- Distanzmatrix: enthält die Kosten der Pfade der Länge 1 zwischen allen Knotenpaaren
- Wie erhält man die Matrix, welche die Kosten eines kürzesten Pfads zwischen je zwei Knoten enthält?

Arten von Bäumen:

- **Shortest Path Tree (SPT):** Verbindet **einen Wurzelknoten** mit jeweils minimalen Kosten mit jedem anderen Knoten des Netzwerks
- **Minimum Spanning Tree (MST):** Verbindet alle Knoten des Netzwerks mit insgesamt minimalen Kosten

Verbindungscharakterisierung (Eigenschaften einer Verbindung zwischen zwei Knoten)

- **Übertragungsrate** r in bit/s bestimmt die notwendige Zeit, um L Datenbits auf ein Übertragungsmedium zu legen (Serialisierungszeit) -> $\boldsymbol{t_s = L/r}$
- **Übertragungsverzögerung**
  - Ausbreitungsverzögerung über eine Distanz d: $\mathbf{t_p = d/vc_0}$
  - Gesamtverzögerung: $\mathbf{t_d = t_s + t_p} = L/r + d/vc_0$
  - **Bandbreitenverzögerungsprodukt** („**Speicherkapazität**“ des Übertragungskanals):
    - Anzahl an Bits (Kapazität), die sich in einer Senderichtung gleichzeitig auf der Leitung befinden können -> $\mathbf{C = t_p r = d/vc_0 \cdot r}$
- **Übertragungsrichtung (Simplex (eine Richtung), Halbduplex (beide Richtungen), Vollduplex)**
- **Mehrfachzugriff (Multiplexing)**: Nachrichten unterschiedlicher Teilnehmer gemeinsam über eine Leitung zu übertragen -> Hub bildet ein **gemeinsames geteiltes Medium**, auf das die Computer mittels eines **nicht-deterministischen Medienzugriffsverfahrens** abwechselnd zugreifen

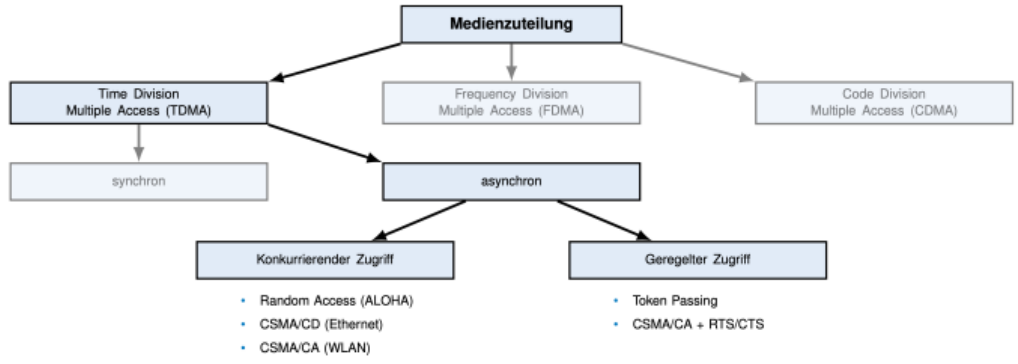
This document is available free of charge on studocu

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

Medienzugriffsverfahren

Bewertungskriterien:

- **Durchsatz:** Nachrichten pro Zeiteinheit
- **Verzögerung** für einzelne Nachrichten
- **Fairness** zwischen Teilnehmern des selben Mediums
- **Implementierungsaufwand** für Sender und Empfänger



TDMA

-> ***synchron*** (**statische** Aufgeteilung, Datenverkehr aber stossartig/burst-artig, Bandbreite steht während Ruhepausen anderen Teilnehmern nicht zur Verfügung)
-> ***asynchron*** (**dynamische** Aufteilung / Zuweisung von Zeitslots, **Zufälliger, konkurrierender oder dynamisch geregelter Medienzugriff**)

Random Access (ALOHA)
1. Jede Station sendet an zentrale Station
2. Senden zwei Stationen gleichzeitig, kommt es zu Kollisionen
3. Erfolgreiche Übertragungen werden quittiert (auf anderer Frequenz, „out-of-band“)
   - Ob ein bestimmter Knoten i innerhalb des Zeitintervalls [t,t + T] zu senden beginnt oder nicht entspricht einem **Bernoulli-Experiment** mit Erfolgs- bzw. Sendewahrscheinlichkeit p_i (Sendewahrscheinlichkeit für alle Knoten gleich)
   - **Binomialverteilung** beschreibt die Anzahl der Erfolge einer Serie gleichartiger und unabhängiger Versuche
   - Für sinnvoll großes N kann die Binomialverteilung durch eine **Poisson-Verteilung** approximiert werden
   - Die mittlere erwartete Anzahl von Nachrichten pro Intervall ist gegeben als **Np = λ**.

Slotted ALOHA -> getaktet

Carrier Sense Multiple Access (**CSMA**)
-> einfache Verbesserung von Slotted ALOHA: „Listen Before Talk“
-> Medium abhören
-> Erst dann senden, wenn Medium frei ist

- 1-persistentes CSMA (Wenn Medium frei, beginne Übertragung)
- p-persistentes CSMA (Wenn Medium frei, übertrage mit Wahrscheinlichkeit p oder verzögere mit Wahrscheinlichkeit 1 – p um eine feste Zeit)
- nicht-persistentes CSMA (Wenn belegt, warte eine zufällig gewählte Zeitspanne)

CSMA/CD (Collision Detection)
- Erkenne Kollisionen und wiederhole die Übertragung, wenn eine Kollision erkannt wird
- Verzichte auf Senden von Bestätigungen
- Wird keine Kollision erkannt, gilt die Übertragung als erfolgreich
- **Problem**: Der Sender muss die **Kollision erkennen**, während er noch überträgt
- Nachrichten müssen folgende **Mindestlänge L_{min}** aufweisen: $L_{min} = 2d/vc_0 \cdot r$, d. h. $t_s >= 2t_p$

-> 1-persistentes CSMA hier Problematisch, nach Collision senden beide wieder => Lösung: Sender wählt zufällig **$n \in \{0, \ldots, \min\{2^{k-1} - 1,1023\}\}$** bei **k-tem Sendeversuch** & wartet n Slotzeiten ==> **Wartezeiten, die situationsabhängig größer** werden (Backoff-slots)

CSMA/CA (Collision Avoidance)
- In Funknetzwerken **funktioniert CSMA/CD nicht**, da der **Sender** einer Nachricht eine Kollision auch bei ausreichender Nachrichtenlänge **nicht immer detektieren** kann.

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

- Contention Window
- CSMA/CA basiert auf p-persistentem CSMA

==> **<u>Erweiterung:</u> RTS/CTS** (Request to Send / Clear to Send) —> **Virtual Carrier Sensing**
Vorteile: Kollisionen werden vermieden (mit oder ohne Hidden Stations), aber nicht gänzlich verhindert
Nachteile: Es können noch immer Kollisionen auftreten, z. B. wenn B das CTS nicht empfängt, TS/CTS nimmt vorab Zeit in Anspruch, was die **maximal erzielbare Datenrate reduziert**

<u>Token Passing</u>

- Kollisionsfreie Übertragung durch Weitergabe eines Tokens
- Stationen werden zu einem logischen Ring zusammen geschaltet, Token zirkuliert im Ring.
- Will eine Station senden, nimmt sie das Token vom Ring und darf danach als einzige Station im Ring übertragen.

-> **Vorteile**:

- Sehr **effizient**, da keine kollisionsbedingten Wiederholungen
- Garantierte maximale Verzögerung (**Determinismus**)

-> **Nachteile**:

- Eine Station muss spezielle Aufgaben übernehmen. (**Monitor-Station**).
- Fehlerhaftes verhalten eines Knotens stört gesamte Kommunikation
- **Übertragungsverzögerung u. U. größer** als bei CSMA, da Sender auf Token warten muss.
- Zusammenschaltung der Stationen zu einem Ring ist u. U. **aufwendig**

Zusammenfassung
-> <u>Konkurrierender</u> Zugriff

- ALOHA und Slotted ALOHA
- CSMA
- CSMA/CD (Kollisionserkennung)
- CSMA/CA (Kollisionsvermeidung)

-> <u>Geregelter</u> Zugriff

- CSMA/CA mit RTS/CTS
- Token Passing (Kollisionsverhinderung)

Rahmenbildung, Adressierung und Fehlererkennung

Rahmenerkennung

-> (**Ziel**: Erhaltung der Codetransparenz (Übertragung beliebiger Zeichenfolgen))
-> Längenangabe der Nutzdaten
-> Steuerzeichen (Start / Ende) -> Bsp. 4B5B-Code: 4 bit Eingabe werden auf 5 bit Ausgabe abgebildet, verbleibende 5-bit Worte ohne Abbildung werden als Steuerzeichen verwendet
-> Begrenzungsfelder und „Bit-Stopfen“
-> Coderegelverletzung

Direktverbindungsnetz
-> alle Hosts sind direkt bzw. über wenige Switches erreichbar (**kein Routing** zwischen Knoten)
-> Broadcast Adresse (spricht alle Knoten im Direktverbindungsnetz an)
-> Multicast Adressen (sprechen bestimmte Gruppe im Netz an)
-> **MAC-Adressen** bieten **keine logische Struktur zur Adressierung** (nur keine Doppelten)
-> Gruppieren (Subnetze) durch MAC-Adressen nicht möglich

MAC-Adressen
-> 6 Byte lang (OUI, Device ID)
-> **Erstes Oktett, letzten zwei Byte: 1. Unicast/Multicast & 2. Globally Unique/Locally Administered**
-> Broadcast: ff:ff:ff:ff:ff:ff

Fehlererkennung
-> Fehlererkennende Codes (Prüfsumme/Checksum), keine Korrektur!
-> **Cyclic Redundancy Check (CRC)**
-> CRC-n Erkennt nicht

- Fehler, die Länger sind als n

This document is available free of charge on studocu

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

- Fehler aus mehreren Bursts
- Fehler, die ein Vielfaches des Reduktionspolynoms sind

Kollisionsdomäne: Teil eines Direktverbindungsnetzes, innerhalb dem eine Kollision bei gleichzeitiger Übertragung mehrerer Knoten auftreten kann, **Switch/Bridge, Router unterbricht** Kollisionsdomäne
Broadcast-domäne: entspricht **Direktverbindungsnetzt**, unterbrochen durch **Router**

Kapitel 3: Vermittlungsschicht

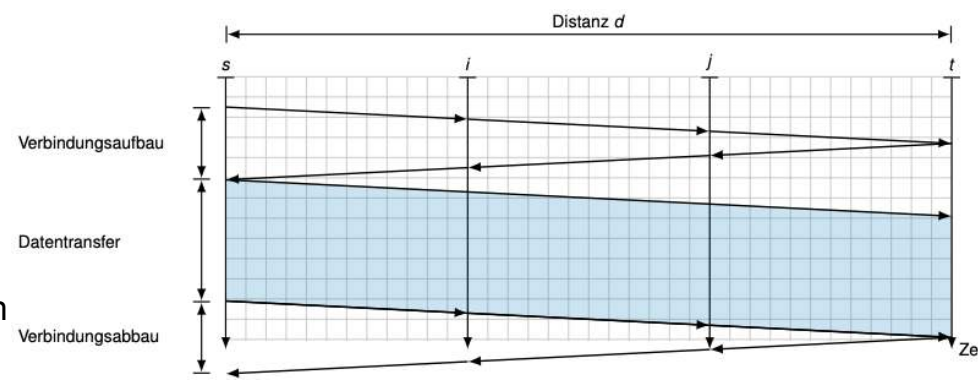
Aufgaben der Vermittlungsschicht:
-> **Kopplung von Direktverbindungsnetzen**
-> **Strukturierte Aufteilung** der kleineren Subnetze
-> **Global eindeutige und strukturierte / logische Adressierung**
-> **Wegwahl** zwischen Geräten über mehrere **Hops**

Vermittlungsarten/Prinzipien für globale Netze
- **Leitungs**vermittlung: **dedizierte Leitung** zwischen Sender und Empfänger, **verbindungsorientiert**
- **Nachrichten**vermittlung: für jede Nachricht als Ganze einen **individuelle Weg**
- **Paket**vermittlung: Teilen der Nachricht in **kleinere Pakete**

1. Leitungsvermittlung (verbindungsorientierte Übertragung)
   (1) **Verbindungsaufbau**: Signalisierungsnachrichten zum Aufbau (beinhaltet Wegwahl, vor Beginn der Datenübertragung)
   (2) **Datenaustausch**: exklusive Nutzung des Kanals durch Kommunikationspartner
   (3) **Verbindungsabbau**: Sigalisierungsnachrichten zum Abbau, Ressourcen werden wieder freigegeben

Vorteile:
- Gleichbleibende Güte der dedizierten Verbindung nach Abbau
- **Schnelle, unkompliezierte Datenübertragung**

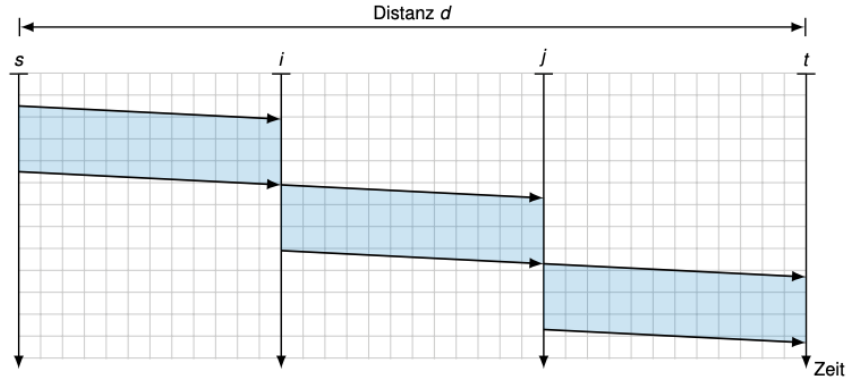
Nachteile:
- Ressourcenverschwendung sofern Leitung nicht dauerhaft ausgelastet, da Leitung **exklusiv reserviert**
- Verbindungsaufbau kann komplex und länger sein
- Hoher Aufwand beim Schalten physikalischer Verbindungen



2. Nachrichtenvermittlung
- Aufbau/Abbau entfallen -> keine Verzögerung
- Header mit Adressinformationen (globally unique?) => **PDU**, als ganzes übertragen
- Möglichkeit für asynchrone Kommunikation
- Mögliche **Zeitersparnis** (Aufbau/Abbau-Phasen entfallen)
- Vermittlungsknoten!!

Vorteile:
- **(Zeit-)Multiplexing** (da keine vorgegebenen Pfade -> gemeinsame Nutzung von Teilstrecken)
- Bessere Ausnutzung der Kanalkapazität
- keine Verzögerung durch Aufbau-Signalisierungsnachrichten

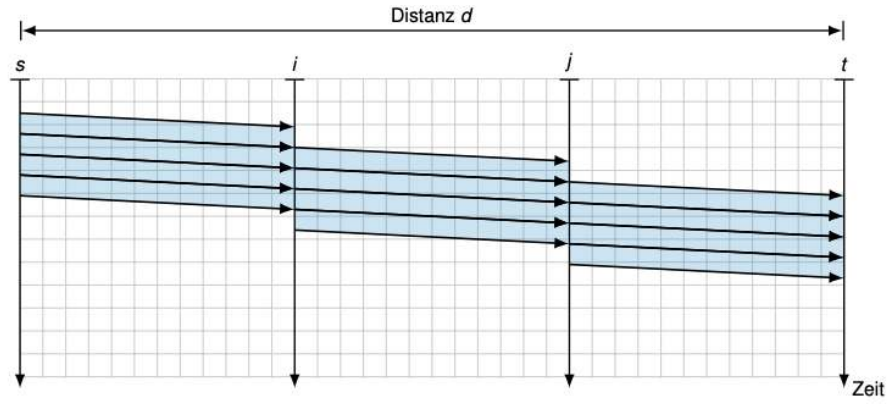
Nachteile:
- Pufferung von Nachrichten, wenn Vermittlungsknoten ausgelastet
- Verlust von Nachrichten möglich
- Mehrfache Serialisierung der ganzen Nachricht



3. Paketvermittlung

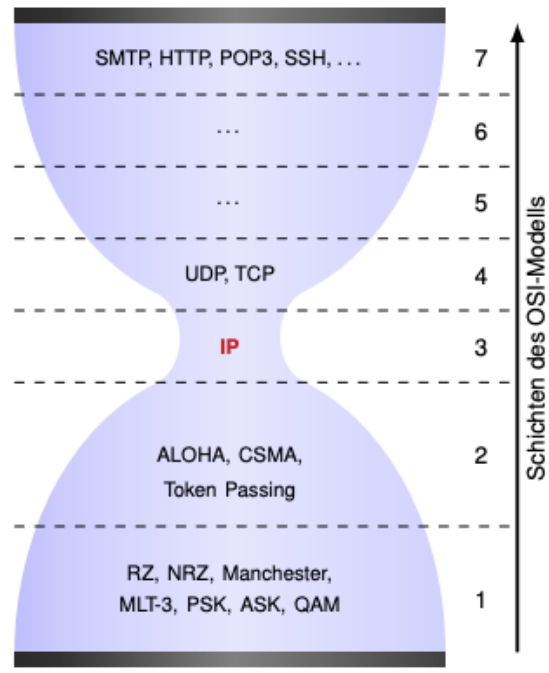
Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

- Jedes Paket bekommt eigenen Header mit Adressinformationen und Reassemblierungs-Infos
- Pakete werden unabhängig voneinander vermittlet (unterschiedliche Wege)
- Pakete müssen nicht gleichgroß sein, aber maximale Paketgröße (p_max)
- Multiplexing -> Engpässe werden fairer genutzt
- Geht ein Packet verlohren, muss nicht die ganze Nachricht wiederholt werden



Vorteile:
- **Zeitmultiplex**
- **Pufferung kleiner Pakete statt ganzer Nachricht (geringere Verzögerung durch Serialisierungszeit)**

Nachteile:
- Verlust von Paketen möglich
- Jedes Paket benötigt eigenen Header (Overhead)
- Empfänger muss Pakete zusammensetzen



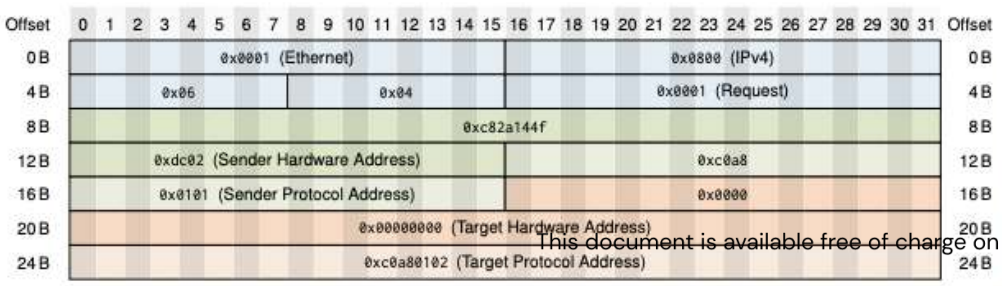
Internet Protocol version 4 (IPv4)

-> IP-Header: Absender-IP-Adresse & Ziel-IP-Adresse (minimale Länge 20B)
- **Version**: 4
- **IHL (IP-Header Length)**: Länge des IP-Header in Vielfachen von **4 Byte** (**Mindestlänge 5** (=**20B**))
- **TOS (Type of Service)**: Klassifizierung und Priorisierung von IP-Paketen
- **Total Length**: Gesamtlänge des IP-Pakets (**Header + Daten**) **in Bytes** (Maximallänge: 65 535B) -> **MTU := maximale Größe eines Pakets auf Schicht 3** inkl. Header /auf Schicht 2 nur Payload (abhängig von Schicht 2/1)
- Identification: zufällig gewählter Wert zur Identifikation zusammengehöreder Fragmente
- **Flags**: **Bit 16** -> 0; **Bit 17** -> Don't Fragment (DF); **Bit 18** -> More Fragments (MF)
- **Fragment Offset**: absolute Position der Daten in diesem Fragment bezogen auf das unfragmentierte Paket in **ganzzahligen Vielfachen von 8 B**
- **TTL**: Leitet ein Router ein IP-Paket weiter, so dekrementiert er das TTL-Feld um 1. Erreicht das TTL-Feld den Wert 0, so verwirft ein Router das Paket und sendet eine Benachrichtigung an den Absender (ICMP Time Exceeded → später). => Maßnahme gegen Routing Loops
- Protocol: Identifiziert das Protokoll auf Schicht 4, welches in der Payload (Datenteil) des IP-Pakets enthalten ist
- Header Checksum: Prüfsumme, welche **nur den IP-Header** (ohne Daten) schützt. Fehlererkennung aber keine Korrektur möglich.

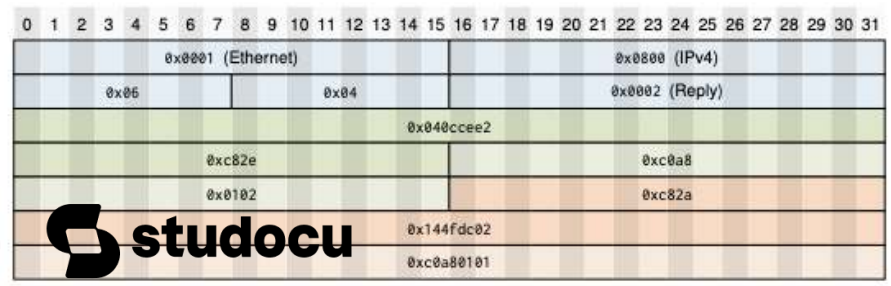
-> Daten (Payload) -> MIND. 1B

Big Endian: „Niederwertigstes Byte steht an höchstwertigster Adresse" (**Network Byte Order**)
Little Endian: „Niederwertigstes Byte steht an niederwertigster Adresse" (meistens Host Byte Order)

Address Resolution Protocol (**ARP**) -> um an MAC-Adresse eines Hosts/Routers zu kommen
1. Host1 sendet einen ARP Request
2. Host2 antwortet mir ARP Reply



(c) ARP Request



(d) ARP Reply

This document is available free of charge on studocu
Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

Default Gateway: Adresse des **Routers** zum Internet, an das alle Pakete geschickt werden, deren Zieladresse **nicht im eigenen Netz** liegen.

M e r k e:

-> **MAC-Adressen** dienen zur **Adressierung innerhalb eines (Direktverbindungs-)Netzes** und werden beim Forwarding verändert

-> **IP-Adressen** dienen der **End-zu-End-Adressierung** zwischen mehreren (Direktverbindungs-)Netzen und werden beim Forwarding **nicht verändert**.

Internet Control Message Protocol (**ICMP**)

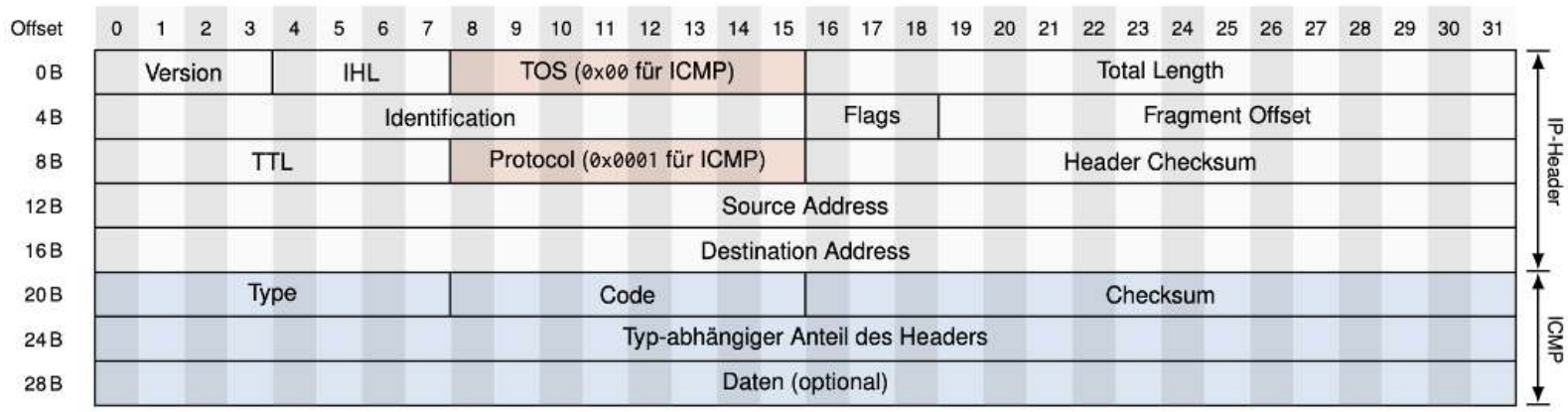


Abbildung 3: Allgemeiner ICMP-Header mit vorangestelltem IP-Header

-> **Fehlerbenachrichtigung** bei
- Pakete in Routing-Schleifen
- Router kennt keinen Weg zum Zielnetz
- der letzte Router zum Ziel kann die MAC-Adresse des Empfängers nicht auflösen
- ICMP Time Exeeded (**Traceroute**: Host sendet z.B. Echo Requests an Host2, setzt TTL Feld zu Beginn auf 1, erhöt TTL schrittweise um 1)

-> Erreichbarkeit von Hosts prüfen (**„Ping“: Echo Request/Reply**)

Dynamic Host Configuration Protocol (**DHCP**) —> **Layer 2 -> UDP**
-> **DHCP-Server weist Hosts IP-Adressen dynamisch zu**
Ablauf:
1. Client sendet **DHCP-Discover** (Layer 2 Broadcast)
2. **DHCP-Offer** (Server bietet Client IP-Adresse an)
3. **DHCP-Request** (Client frägt IP-Adresse an)
4. **DHCP-ACK** (Server gibt IP-Adresse frei) oder **DHCP-NACK** (Nutzung wird untersagt)

-> Begrenzte Gültigkeit/Lease

=> Subnetting (Classless Routing)
-> Verfahren zur Unterteilung von IP-Netzen: Subnetzmaske -> unterteilt IP-Adresse in Netz- (1) und Hostanteil (0)
-> IP-Adresse **AND** Subnetzmaske => Netzadresse

IPv4 Adresslänge: 32bit
Erste IP (Netzadresse)
Letzte IP (Broadcast-Adresse)

Anzahl IP-Adressen eines Adressbereichs (X=Präfix): 2^{32-X} (Netzadresse & Broadcast-Adresse nicht für Hostadressierung verfügbar)

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

-> Adressblöcken in einem einzigen **Subnetz zusammenzufassen** geht nur, wenn
1. Subnetze **gleich groß** sind (**Achtung**: Kriterium ist notwendig, nicht hinreichend!).
2. wenn sie **benachbart sind** und
3. sich **im nächst größeren Subnetz** zusammen fassen lassen. (Letzteres Kriterium ist gleichbedeutend mit einem gemeinsamen Vaterknoten der beiden Subnetze, wenn man sich den Adressraum als Binärbaum vorstellt.)

=> Besondere Adressbereiche IPv4:
1. **0.0.0.0 / 8**: Hosts in diesem Netzwerk (Adressen aus diesem Bereich werden grundsätzlich **nicht geroutet**.)
2. **127.0.0.0 / 8**: „Loopback-Adressen“ (identifizieren den lokalen Rechner (Localhost), Pakete bleiben im Host, grundsätzlich **nicht geroutet**)
3. **10.0.0.0 / 8**, **172.16.0.0 / 12**, **192.168.0.0 / 16**: Private Adressbereiche (dürfen zwischen **privaten Netzen geroutet** werden)
4. **169.254.0.0 / 16**: Automatic Private IP Adressing (APIPA) (Block zur automatischen Adressvergabe, Routing nur in privaten Netzen)
5. **255.255.255.255 / 32**: Global Broadcast (Identifiziert im Prinzip alle Hosts, wird niemals geroutet)

Aufteilung in Subnetzte
Größe = 2* Bedarf
Nuzbar = Größe - 2
Präfixnotation: normal, aber wenn durch 2, dann Präfix + 1
Subnetzmaske: 1111.1111.1110.0000 => 255.255.2xx.xxx
Netzadresse: 1. Adresse
Broadcast-Adresse: letzte Adresse

Internet Protocol version 6 (IPv6)

-> **Vergrößerter** Adressraum: 2128
-> Vereinfachung des **Headerformats**
-> **IP-Fagmentierung** übernimmt der Host, nicht mehr der Router
-> Flexibilität durch sog. **Extension Header**
-> Stateless Address Autoconfiguration (**SLAAC**) mittels ICMPv6.
-> Möglichkeit für **Stateful Autoconfiguration** durch DHCPv6.
-> Nativer Einsatz von **Multicast**, beispielsweise um alle Router in einem Segment zu adressieren

Aufbau:
1. Prefix
2. Subnet-Identifier
3. Interface Identifier

-> kompakte Schreibweise
- führende Nullen werden weg gelassen: 2001:db8:1:d:0:f0:0:0
- der größte konsekutive Block von mindesten 2 „Nuller“-Blöcken kann durch :: abgekürzt werden: 2001:db8:1:d:0:f0::
- Beispiel Loopback-Adresse: **::1/128**

Adresslänge IPv6: 128 bit
Anzahl an Adressen: 2^{128}-64

Header (Mindestlänge 40B):
- **Traffic Class**: Äquivalent zum TOS-Feld des IPv4-Headers, Wird zur Verkehrspriorisierung / Quality of Service (QoS) verwendet
- **Flow Label**: von Routern verwendet, um zusammengehörende Pakete (Flows) auf Schicht 3 zu erkennen
- **Payload Length**: (inkl. Extension-Header) in **Bytes**,

This document is available free of charge on studocu

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

- **Next Header**: Typ des nächsten Headers (L4-Header (TCP, UDP, …), ICMPv6-Header oder IPv6 Extension Header)
- **Hop Limit**: TTL
- **Extension Header**: Layer 3 Informationen, werden **nur vom Empfänger ausgewertet** (Ausnahme Hop-by-Hop Options Header)
  Bsp: Fragment-Header

MTU für anderes Segment bestimmen:
-> Packet mit MTU des lokalen Abschnitt und DF(Don't Fragment)-Bit setzen
-> 2 verwirft Nachricht und sendet ICMP-Nachricht Typ 3 Code 4 (Destination Unreachable Fragmentation Needed, DF set) zurück <= enthält maximale MTU für Abschnitt 2

Fragment-Header
- Fragment Offset: Offset der fragmenierten L3-SDU in Vielfachen von **8 B** (Fragmentierung und Reassemblierung nur bei Sender bzw. Empfänger!)
- More Fragments (MF)
- Identification

=> Besondere Adressbereiche IPv6:

1. **::/128** - nicht-spezifizierte Adresse (grundsätzlich **nicht geroutet**)
4. **::1/128** - **Loopback**-Adressen (Localhost (Pakete bleiben im Host), **nicht geroutet**)
5. **fe80::/10** - **Link-Local** Adressen (jedes IPv6 Interface benötigt eine LL-Adresse, gültig nur innerhalb des lokalen Links, **nicht geroutet**) —> 1111.1110.10…
6. **fc00::/7** - **Unique-Local**-Adressen (nur für lokale Kommunikation, **nur lokal geroutet**) —> 1111.101…
7. **ff00::/8** - **Multicast**-Adressen (bestimmte Gruppe von Hosts, **geroutet**) —> 1111.1111…
8. **Rest:** Global Adressen (globally unique, **geroutet**)

Adressierungsarten:

1. **Unicast**: an einzelnes Ziel adressiert
2. **Broadcast**: an alle Stationen im Netzwerk adressiert
3. **Multicast**: an Gruppe von Knoten (Hosts, Router, etc.) adressiert
   - All Nodes -> ff02::1
   - All Routers -> ff02::2
   - All DHCP-Agents -> ff02::1:2
   - Solicited-Node Address -> Neighbor Discovery Protocol -> ff02::1:ff00:0/104 + letzte 24 bit der IPv6-Adresse
4. **Anycast**: an beliebige Station einer bestimmten Gruppe adressiert

***Mapping** von Multicast IPv6 Adressen auf MAC-Adressen*

1. Die ersten 2 Oktette der MAC-Adresse werden auf 33:33 gesetzt
   - **letztes** Bit des ersten Oktetts ist gesetzt → **Multicast**
   - **vorletztes** Bit des ersten Oktetts ist gesetzt → **locally administered (bei IPv6 andersrum)**
2. Die letzten 4 Oktette der Ethernetadresse werden die letzten 4 Oktette der IPv6 Multicastadresse

Stateless Address Autoconfiguration (SLAAC):

-> Host generiert sich **link-local IPv6-Adresse** (für Interface) aus MAC-Adresse
-> Das **Präfix ist fe80::/10**
-> Der **Subnet Identifier** (die folgenden 54 bit) werden **auf 0 gesetzt**
-> Die verbleibenden 64 bit stellen den Interface Identifier dar, welcher aus der MAC-Adresse des jeweiligen Interfaces als modifizierter EUI-64 Identifier generiert wird:
   - Die ersten **24 bit** sind der **OUI** der MAC-Adresse (**vorletztes Bit des 1. Oktetts** des OUI invertiert)
   - Die nachfolgenden **16 bit** werden mit **ff:fe** „gestopft"
   - Die restlichen **24 bit** werden mit dem **Device Identifier** der MAC-Adresse aufgefüllt

-> **Globale (Unique) Adressen** über SLAAC konfigurieren

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

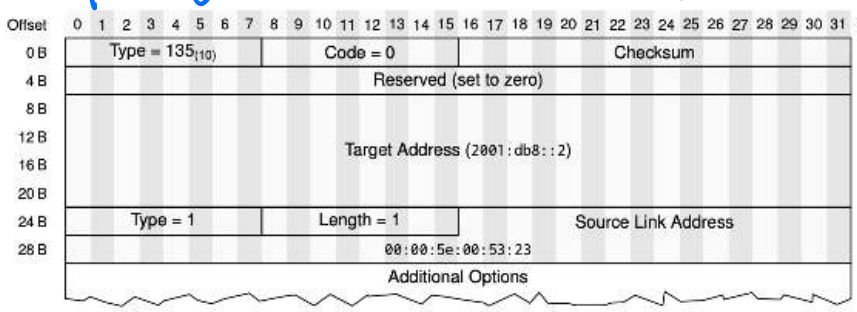
-> **Präfix Informationen** müssen von den Routern über das **Neighbor Discovery Protocol** in Form von **Router Advertisements** an Hosts versendet werden
-> Host kann über das /64 Präfix und dem modifizierten EUI-64 Identifier selbstständig eine Adresse erzeugen
-> **stateless**, da die Adressen nicht von einem Server vergeben werden (wie bei DHCPv6)

==> **Privacy** in Gefahr, da MAC Adresse eindeutig und in einer globalen IP-Adresse eingebettet verfolgt werden kann. **LÖSUNG**: **IPv6 Privacy Extension**

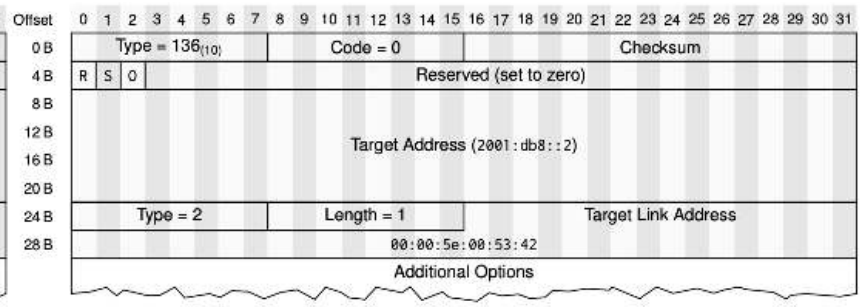
ICMPv6
-> ersten 32 bit sind immer vorhanden
-> Type Feld (Echo, Time Exceeded, …)
-> Code (Subtype: Request/Response)
-> fehlererkennende Checksumme
-> **Neighbor Discovery Protocol (NDP)**

- **Adressauflösung**, **Duplicate Address Detection** und **Neighbor Unreachability Detection**: **Neighbor Solicitations und Advertisements**
  - Neighbor Solicitation (Request) -> **MULTICAST** (**Solicited-Node Adress(Multicast)!!, gemappte MAC-Adr.**)
  - Neighbor Advertisement (Reply) -> UNICAST
    - **Router-Flag R**: antwortender Knoten ist ein Router
    - **Solicited Flag S**: Advertisment infolge einer Solicitation
    - **Override Flag O**: Link-Layer Adresse soll aktualisiert werden

Request



Reply



- Automatisches Auffinden von Routern innerhalb des lokalen Netzsegments, Adress-Präfixen und Parameter Konfiguration: **Router Discovery und Router Advertisements**
- Umleitung zu anderen Gateways: **Redirects**

ROUTING:
-> Longest Prefix Matching
1. R1 berechnet das **logische AND** aus der Zieladresse des Pakets und den Subnetzmasken
2. Vergleich mit Destination Feld
3. Falls gleich, **Bestimmen von Gateway (next Hop) und zugehörigem Interface**
4. **Auflösen der MAC-Adresse des Gateways (next Hop)** (ggf. via ARP), Paket mit einem neuen Ethernet-Header versehen und weiterleiten

DYNAMISCHES ROUTING:
-> Routing-Protokolle: Router tauschen sich aus

- **Distanz-Vektor-Protokolle**:
  - Router kennen nur Richtung (NextHop) und Entfernung (Kosten) zu einem Ziel
  - Router tauschen untereinander lediglich kumulierte Kosten aus
  - Algorithmus von **Bellman-Ford** (einfache Implementierung, Keine komplexen Datenstrukturen notwendig)
- **Link-State-Protokolle**
  - Router informieren einander zusätzlich auch darüber, wie ein Ziel erreichbar ist
  - Komplexe Nachbarschaftsbeziehungen und Update-Nachrichten
  - => vollständige Topologieinformationen
  - **Dijkstras Algorithmus** (Resourcenintensiver, da komplexere Datenstrukturen, bessere Laufzeit)
  - Bsp: OSPF (Open Shortest Path First) (Industriestandard)

-> Routing Information Protocol (RIP)

This document is available free of charge on studocu

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

- Distanz-Vektor-Protokoll
- Einzige Metrik: Hop Count (entspricht Bellman-Ford mit Kantengewicht 1 auf allen Kanten)
- Hop Count Limit von 15
- **PROBLEM 1**: Verzögerung von maximalen Hops * Zeit zwischen Senden (15 * 30s = 7,5 min)
- **LÖSUNG**: **Triggered Updates**: Sobald ein Router eine Änderung an seiner Routingtabelle vornimmt, sendet er sofort ein Update (belastet während Updates aber stark das Netzwerk)
- **PROBLEM 2: Count to Infinity-Effect**: Wenn ein Link ausfällt, wird dieser trotzdem fälschlicherweise weiterverbreitet (bis zu Hop Count Limit), da die Reihenfolge der Updates random ist.
- **LÖSUNGS**-Ansätze:
    - **Split Horizon** („Sende dem Nachbarn, von dem Du die Route zu X gelernt hast, keine Route zu X.“)
    - **Poison Reverse** (Wenn ein Link ausfällt, wird die Route mit unendlicher Metrik gesendet)
    - **Path Vector** (Sendet bei Updates den vollständigen Pfad, über den das Ziel erreicht wird)

Autonomes System (AS)

- **Menge von Netzwerken, die unter einheitlicher administrativer Kontrolle stehen**
- Identifiziert durch Identifier, der sog. **AS-Nummer**
- Innerhalb eines autonomen Systems: **Interior Gateway Protocols (IGPs)** wie **RIP**, **OSPF**, …
- Austausch von Routen zwischen Autonomen Systemen: **Exterior Gateway Protocol (EGP)** -> **Border Gateway Protocol (BGP)**
- Autonome Systeme können durch **Upstream-Provider(C2P)** oder durch **Peering** miteinander verbunden sein

Bedingungen, um zwei Subnetzte zu aggregieren (IPv6):
1. gleich groß (selbe Präfixlänge)
2. benachbart (auf die letzte Adresse im ersten Netz muss direkt das nächste Netz folgen)
3. Es muss eine valide Präfixmaske mit Länge n – 1 existieren, d. h. die beiden Netze dürfen sich nur genau im letzten Bit ihres Präfix unterscheiden.

Bit-Stopfen:
-> Das Verhindern des Auftretens von Steuerzeichen durch das gezielte Einfügen von zusätzlichen Bits in die Payload.

Wir sollten nun wissen,

- was die Unterschiede zwischen Leitungs-, Nachrichten- und Paketvermittlung sind,
- worin der technische und logische Unterschied zwischen den Adressen auf Schicht 2 und 3 besteht,
- welche Möglichkeiten IPv4 und IPv6 zur logischen Strukturierung bieten und wie dies funktioniert,
- wie zu einer gegebenen Adresse auf Schicht 3 die zugehörige Link-Layer Adresse bestimmt wird,
- was eine Routing Tabelle ist,
- wie Router Weiterleitungsentscheidungen treffen,
- was der Unterschied zwischen Routing und Forwarding ist,
- wie Router untereinander Routen austauschen
- welche Arten von Routingprotokollen es gibt und
- wie diese funktionieren.

Kapitel 4: Transportschicht

Aufgaben:
-> **Multiplexing** von Datenströmen unterschiedlicher Anwendungen bzw. Anwendungsinstanzen (Segmentierung der Datenströme unterschiedlicher Anwendungen)
-> Bereitstellung verbindungsloser und verbindungsorientierter **Transportmechanismen**
    - **Verbindunglos** (Best Effort): State-less, Keine Sequenznummern, keine Übertragungswiederholung, keine Garantie der richtigen Reihenfolge (sendto() und recvfrom())

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

- **Verbindungsorientiert**: State-verwaltet, Übertragungswiederholung bei Fehlern, Garantie der richtigen Reihenfolge (Systemaufrufe read() und write())

-> Mechanismen zur Stau- und Flusskontrolle

- **Congestion Control**: Reaktion auf drohende Überlast im Netz
- **Flow Control:** Laststeuerung durch den Empfänger

-> Header entält Quellport & Zielport

-> Anwedung eindeutig identifiziert durch 5-Tupel ⇒ (SrcIPAddr, SrcPort, DstIPAddr, DstPort, Protocol)

-> **Betriebssysteme verwenden 5-Tupel, um Anwendungen Sockets bereitzustellen**

1. VERBINDUNGSLOSE ÜBERTRAGUNG

—> User Datagram Protocol (UDP)

- ungesicherte und nachrichtenorientierte Übertragung bei geringem Overhead
- UDP-Header
  - Lenth von Header + Daten in Bytes
  - Prüfsumme von Header & Daten
- Vorteile:
  - **Geringer Overhead**
  - **Keine Verzögerung** durch Verbindungsaufbau oder Retransmits und Reordering von Segmenten
  - Gut geeignet für **Echtzeitanwendungen!!** (Voice over IP, Online-Spiele)
  - Keine Beeinflussung der Datenrate durch Fluss- und Staukontrollmechanismen
- Nachteile:
  - Keine Zusicherung von Dienstqualität (beliebig hohe Fehlerrate)
  - Datagramme können out-of-order ausgeliefert werden
  - Keine Flusskontrolle & Staukontrollmechanismen
- Bsp.:
  - Namesauflösung mittels Domain Name System (DNS)
  - Datenverkehr mit Echtzeitanforderungen
  - Google's QUIC

2. VERBINDUNGORIENTIERTE ÜBERTRAGUNG

—> Linear durchnummerierte Segmente mittels **Sequenznummern** im Protokollheader

- Sequenznr. ermöglicht
  - **Bestätigung** erfolgreich übertragener Segmente (Forward **ACK**)
  - **Identifikation** fehlender Segmente
  - **Erneutes Anfordern** fehlender Segmente
  - Zusammensetzen der Segmente in der **richtigen Reihenfolge**
- **Problem:** Sender und Empfänger müssen sich zunächst **synchronisieren** (Austausch der initialen Sequenznummern) und Zustand halten
- Verbindungsphasen:
  - Verbindungsaufbau (3-Way-Handshake)
  - Datenübertragung
  - Verbindungsabbau (Teardown)

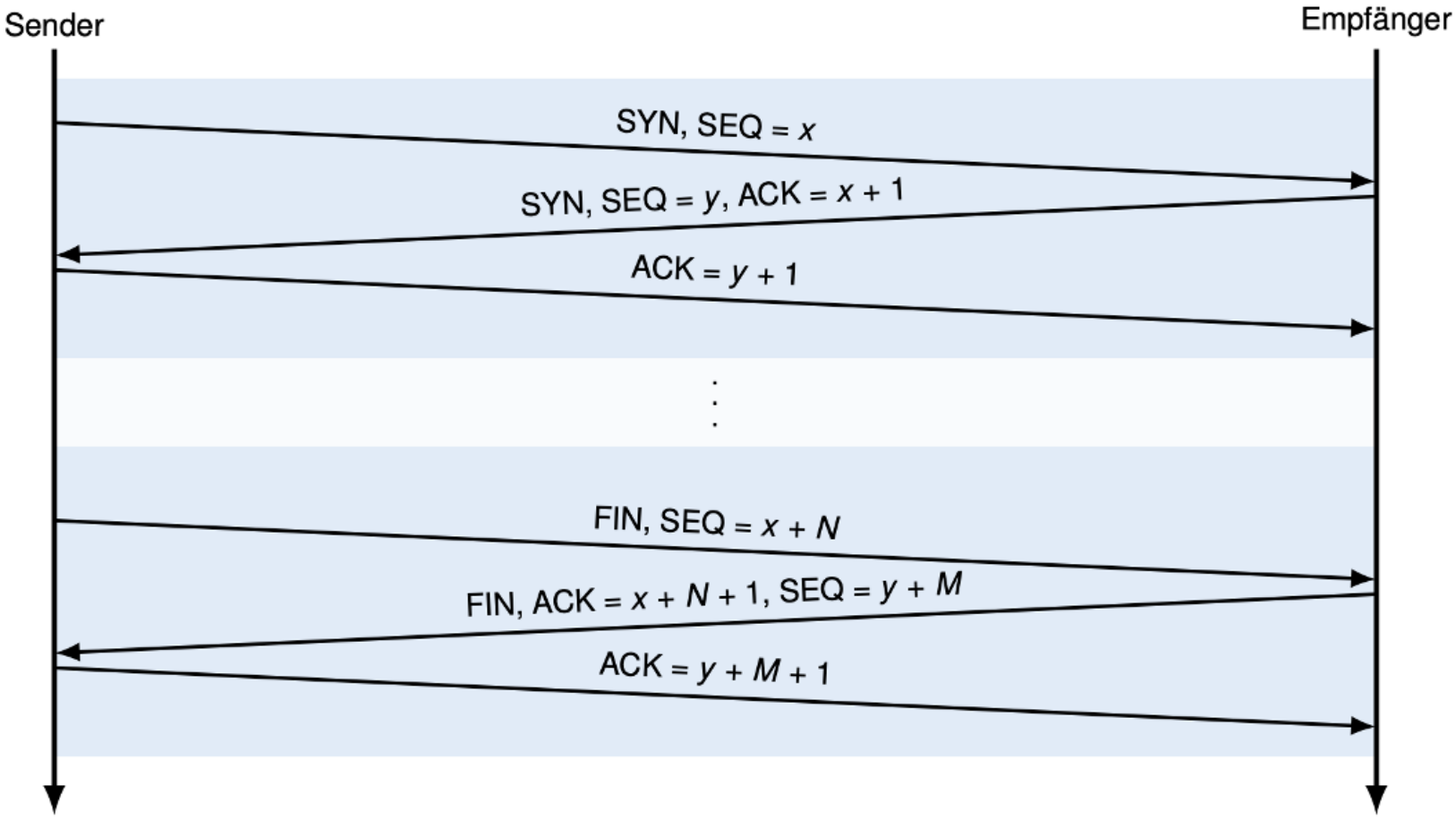
-> **Sliding-Window-Verfahren:**

- Empfänger teilt dem Sender mit, wie viele Segmente nach dem letzten bestätigten Segment auf einmal übertragen werden dürfen, ohne dass der Sender auf eine Bestätigung warten muss
- Vorteile:
  - Zeit zwischen dem Absenden eines Segments und dem Eintreffen einer Bestätigung kann effizienter genutzt werden
  - Durch die Aushandlung dieser **Fenstergrößen** kann der Empfänger die Datenrate steuern → **Flusskontrolle**
  - Durch **algorithmische Anpassung** der **Fenstergröße** kann die Datenrate an die verfügbare Datenrate auf dem Übertragungspfad zwischen Sender und Empfänger angepasst werden → **Staukontrolle**
- -> Sender und Empfänger müssen mehr Zustand halten
- Problem: Segmentverluste
  1. **Go-Back-N**
     - Nur die nächste erwartete Sequenznummer wird akzeptiert, alle anderen Segmente werden verworfen

This document is available free of charge on studocu

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

- Bei einem Sequenznummernraum der Kardinalität N muss für das Sendefenster stets gelten: $w_s \leq N - 1$
- **Einfach zu implementieren aber weniger effizient**



2. **Selective Repeat**
   - Empfänger akzeptiert alle Sequenznummern, die in das aktuelle Empfangsfenster fallen
   - Leistungsfähiger
   - Bei einem Sequenznummernraum der Kardinalität N muss für das Sendefenster stets gelten: $w_s \leq \lfloor \frac{N}{2} \rfloor$

—> Transmission Control Protocol (TCP)

- gesicherte / stromorientierte Übertragung mittels **Sliding-Window** und **Selective Repeat**
- Es werden bei TCP nicht ganze Segmente sondern einzelne Bytes bestätigt (**stromorientierte Übertragung**) —> Die Acknowledgement-Number gibt bei TCP stets das **nächste erwartete Byte** an
- **(Data) Offset**: Länge des TCP-Headers in Vielfachen von **4B**
- Flags:
  - **URG**: Daten im aktuellen TCP-Segment beginnend mit dem ersten Byte bis zu der Stelle, an die das Feld Urgent Pointer zeigt, sofort an höhere Schichten weitergeleitet
  - **ACK**: **Empfangsbestätigung**
  - **PSH**: Sende- und empfangsseitige **Puffer des TCP-Stacks umgangen**
  - **RST**: TCP-**Verbindungsabbruch** ohne ordnungsgemäßen Verbindungsabbau
  - **SYN**: Segment, welches zum **Verbindungsaufbau** gehört
  - **FIN**: Segment, welches zum **Verbindungsabbau** gehört
- Receive Window: Größe des aktuellen Empfangsfensters in Byte, Kontrolle über Datenrate
  - Maximale Anzahl an Byte, die ohne Abwarten einer Bestätigung übertragen werden dürfen
- Checksum: Prüfsumme über Header und Daten
- Urgent Pointer: Gibt das Ende der „Urgent-Daten“ an
- Options: Window Scaling, Selektive Bestätigung, **Maximum Segment Size (MSS)** -> maximale Größe eines TCP-Segments(Nutzdaten)
  - Beispiel: MSS = MTU - 20B (TCP-Header) - 20B (IP-Header) (- 8B (PPPoE-Header falls DSL))
- **Staukontrollfenster** (engl. Congestion Window) W_c -> wird verkleinert, wenn Verluste auftreten, für das tatsächliche Sendefenster gilt stets $\mathbf{w_s = min\{w_c, w_r\}}$
- Zwei Phasen der Staukontrolle:
  - Slow Start (exponentielles Wachstum bis Congestion Threshold)
  - Congestion Avoidance (lineares Wachstum)

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

NETWORK ADDRESS TRANSLATION (NAT)

—> Technik zur Übersetzung von N ≥ 1 auf M ≥ 1 andere IP-Adressen (Grund: **Adressknappheit bei IPv4**)
—> Bei IPv4: Abbildung von N **privaten (nicht öffentlichen)** auf M **öffentliche (global eindeutige)** IP-Adressen
—> NAT ist **keine Firewall!** -> NAT kann zwar grundlegenden Schutz bieten (erlaubt keine eingehende Verbindungen ohne vorherigen Verbindungsaufbau), aber **wurde nicht aus Sicherheitsgründen implementiert und erfüllt nicht die Funktionen einer Firewall**.
—> **IPv6: Präfix Übersetzung**

- Private IPv6-Adressen können **in global gültige Adressen übersetzt werden** (ohne Ports oder Identifier, zustandslos, ohne NAT-Tabelle) & **man kann Host-Adresse damit verbergen**

Kapitel 5: Sitzungschicht, Darstellungsschicht, Anwendungsschicht

SITZUNGSSCHICHT

-> Verbindungsorientiert

- Verbindung zwischen den Kommunikationspartnern
- Verbindung bleibt über Dauer einzelner Transfers der Transportschicht bestehen
- 3 Phasen: Verbindungsaufbau, Datentransfer und Verbindungsabbau

-> Verbindungslos

- Daten werden an die Transportschicht durchgereicht

-> **Session**: Kommunikation zwischen mindestens zwei Teilnehmern mit definiertem Anfang und Ende sowie sich daraus ergebender Dauer (kann mehrere Transportschicht-Verbindungen beinhalten)

-> HTTP:

- ursprünglich zustandslos
- Cookies
- kleine Datenfragmente, welche von Webserver an Client übertragen und dort gespeichert werden, um Anfragen eines bestimmten Clients/Nutzers wiedererkennen zu können
- ermöglichen Sitzungen/Sessions über mehrere TCP-Verbindungen

-> TLS:

- Protokoll zur Authentifizierung und verschlüsselten Übertragung von Daten über einen verbindungsorientierten Transportdienst
- Grundlage für HTTPS
- Authentifizierung (Sitzungsschicht), Vertrauligkeit(Verschlüsselung) und Integritätsschutz (Darstellungsschicht)
- Sitzungen können mit Session-IDs über mehrere TCP-Verbindungen erhalten bleiben (Sitzungsschicht)

DARSTELLUNGSSCHICHT

—> Ermöglicht den Kommunikationspartnern einheitliche Interpretation der Daten (Übertragung der Daten in einheitlichem Format)
—> Aufgaben:

- **Darstellung der Daten**
- **Datenstrukturen zur Übertragung**
- **Darstellung der Aktionen an diesen Datenstrukturen**
- **Datentransformationen**
- **Anwendungen sollen syntaxunabhängig miteinander kommunizieren können**

—> Funktionen:

- Kodierung der Daten (Übersetzung zwischen Zeichensätzen und Codewörtern, Kompression, Verschlüsselung)
  - Fixed-Length Codes (Codewörtern derselben Länge) -> ASCII

This document is available free of charge on studocu

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

        - Variable-Length Codes (Codewörtern unterschiedlicher Länge) -> UTF-8
    - Strukturierte Darstellung von Daten (plattformunabängig, einheitlich, Serialisierung)
        - (gepackte) structs / serialisierte Speicherbereiche
        - Ad-hoc Datenformate
        - Strukturierte Serialisierungsformate wie **XML oder JSON**

—> Datenformen:
- Textzeichen bzw. Symbole in lesbarer Form
- Binäre Daten

—> **Datenkompression -> Entfernung von Redundanz**

1. **Verlustfreie** Komprimierung (ZIP, PNG, FLAC)
    - **Huffman-Code** (statisch): häufigen Zeichen kürzere Codewörter zugewiesen, Abbildung zwischen Zeichen und Codewörtern bleibt dabei eindeutig und umkehrbar, **optimaler** und **präfixfreier** Code
        - **Präfixcode**: **Codewörter sind niemals Präfix eines anderen Codeworts** ==> Daten können sofort rekonstruiert werden, wenn ein Codewort gefunden wurde
        - **Optimaler Code: minimiert die mittlere Codewortlänge**
    - **Run-length Codes**: wenn Zeichen wiederholt vorkommen, wird die Anzahl der Wiederholungen kodiert (**nicht optimal**)
2. **Verlustbehaftete** Komprimierung (MP3, MPEG, JPEG)

ANWENDUNGSSCHICHT

—> **Domain Name System (DNS)**
- Auflösung von Domänennamen in IP-Adressen und umgekehrt
- **Domaine Namespace**: hierarchisch aufgebauter Namensraum, verteilte Datenbank
- **Nameserver**: Speichert Informationen über Domains im Namensraum
    - Nameserver erwarten eingehende Verbindungen auf **Port UDP/TCP 53**
    - **Zone:** zusammenhängende Teilbäume des Namespaces
    - Nameserver ist **autoritativ** für die Zonen, die er speichert
    - **Glue Record**: enthält IP-Adresse für gesuchten autoritativen Nameserver
- **Resolver**: **Namensauflösung**: Server, die Informationen aus dem DNS extrahieren und das Ergebnis an den anfragenden Client zurückliefern
    - **Rekursive** DNS-Anfragen: Resorver leitet Namensauflösungs-Anfrage an anderen Resolver weiter
    - **Iterative** Anfrage: Resolver erledigt eigentliche Namensauflösung mittels einer Reihe von Anfragen an autoritative Nameserver
- Fully Qualified Domain Name (**FQDN**): Startet an Knoten des Namespaces, endet an Wurzel (.)
- Top Level Domain (**TLD**): com, de, net, …
- Second Level Domain (**SLD**): google, tum, reddit, …
- PTR Records: Reverse DNS (IP-Adr. —> FQDN)

Reverse DNS: Zu IP-Adressen können FQDNs hinterlegt werden
-> mittles **PTR Records**
-> Zonen: **in-addr.arpa. (IPv4**, jede Ebene ein Oktett**)** und **ip6.arpa. (IPv6 jede Ebene 4 bit)**

—> Uniform Resource Locator (URL)
- **Adressangaben, um das Anwendungsprotokoll anzugeben & bestimmte Resource adressieren zu können**

—> HyperText Transfer Protocol (HTTP)
- **Transfer von Webseiten und Daten**
- **definiert, welche Anfragen ein Client stellen darf und wie Server darauf zu antworten haben**
- Port TCP 80
- Request und Response

—> File Tranfer Protokoll (FTP)
- **Transfer von Dateien von und zu Servern**

—> Simple Mail Transfer Protocol (SMTP)
- **Versandt von Emails, Kommunikation zwischen Mailservern**

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)

—> Internet Message Access Protocol (IMAP) & POP
    - **Protokolle, mit denen ein Mail User Agent auf zugestellte Emails zugreifen kann**
—> Telnet
    - **Interaktive Kommunikation mit anderem Host**
—> Secure Shell (SSH)
    - **Verschlüsselte entfernte Anmeldung an einem Host**
—> Simple Network Management Protocol (SNMP): **Monitoring & management von Netzkomponenten**

This document is available free of charge on studocu

Downloaded by ?? ? (tm7vs669np@privaterelay.appleid.com)