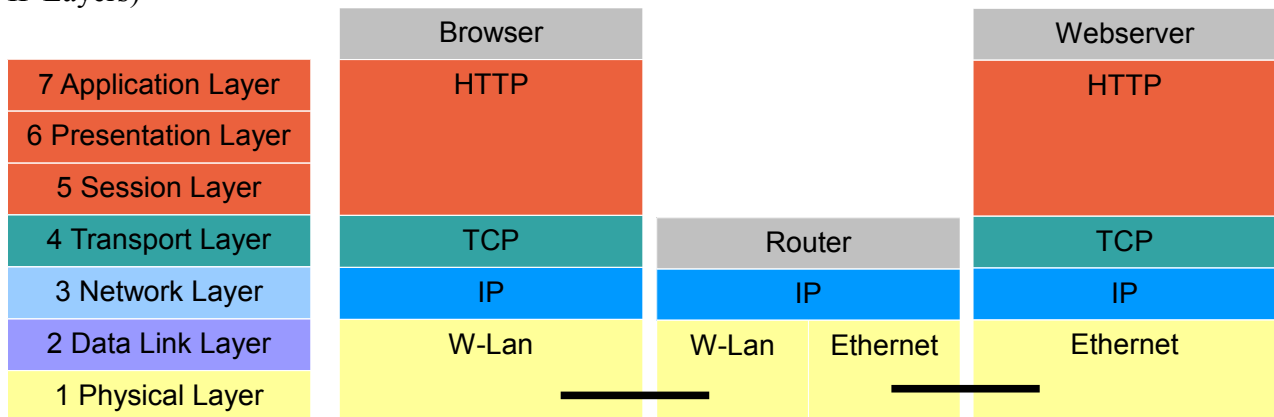


## OSI Open System Interconnect

OSI ist nur ein Referenzmodell. In der Realität sind einige Layer zusammengefasst. (Ethernet Stack)  
Weiter unten liegende Layer bieten den höheren Layer Services an. So braucht der TCP Service nicht zu wissen, in welchem Format die Daten versant werden. (zB Wireless oder Ethernet)

In jedem Layer werden dem vom letzten Layer übergebenen Package ein Header hinzugefügt, damit dieses beim Empfänger wieder entpackt werden kann. Somit sind beim Empfänger die selbigen Layers von nöten wie beim Sender. Peer-to-Peer gesehen sind die Layers somit gleichgestellt. (zB IP Layers)



Anwendungsschichten (5-7) sind in der Realität oft kaum zu unterscheiden. Sind für das Zusammenwirken mit dem Programm und OS verantwortlich.

## Übertragungsmedium

Nicht im OSI Referenzmodell

*Wichtigste Eigenschaften:* Erreichbare Distanz, maximale Datenübertragungsrate

Durch (aufwändige) Singalverarbeitung kann viel Leistung gewonnen werden.

**-Leitungscode** Bandbreite effizient nutzen, Taktrückgewinnung, Gleichspannungsfrei

**AMI** (Alternate Mark Inversion)

3 Zustände, +/0/-, 1 werden

abwechslungsweise + und -

Problem: Takt bei langen 0-Folgen

HDB3	# Pulse seit letzter Ersetzung	
Letzer Puls	ungerade	gerade
Positiv	000+	-00-
Negativ	000-	+00+

**Manchester Codierung:** 1=positive-,

0=negative Flanke, einfache

Taktrückgewinnung 10Base-T

**NRZI** Non-Return-to-Zero Invert on 1

Bei einer Null bleibt der

Ausgangspegel wie er ist, bei einer

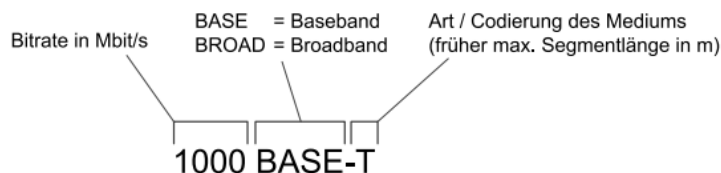
Eins erfolgt ein Pegelwechsel.

Typischerweise bei 100Base-TX

**4B5B** 4 Datenbits auf 5 Gesendete

Bits für Takt (Keine langen 0-folgen)

Wird mit NRZI verwendet



## Signalämpfung / Leistungsabnahme

Wesentlicher Faktor der erreichbaren Distanz. Verstärkung des Signals beim Empfänger möglich, aber ab einer gewissen Distanz ist die Störung stärker als die Distanz.

Mögliche Ursachen für die Störung: Übersprechen von anderen Leitungen, Rauschen des Empfängers, elektromagnetische Einstreuung von anderen Quellen.

## Koaxialkabel

Geringe Dämpfung. Punkt-Punkt Verbindungen (10-100 Gigabit) & Bus Netzwerke leicht zu realisieren.

Anfällig auf Schäden (Quetschen & Knicken)

*Paarsymmetrische Kabel (Twisted Pair)* Twisted Pair [Xcrossed] bsp: 100 Base-TX

Shielded und Unshielded. Shielded verbessert Störsicherheit, sind aber teurer, schwerer und steifer. Schild kann aus Drahtgeflecht, metall beschichteter Folie oder beidem bestehen.

Normalerweise 4 Aderpaare (8 Kabel)

Twisted Pair Namensgebung

xx/yTP

bsp. SF/UTP

xx	Gesamtschirmung	y	Aderpaarschirmung
U	ungeschirmt	U	ungeschirmt
F	Folienschirm	F	Folienschirm
S	Geflechtsschirm	S	Geflechtsschirm
SF	S & F kombiniert.		

Paarsymmetrische Kabel haben eine Passive (Empfänger) Leitung und eine Aktive (Sender) Leitung. Es wird ein Komplementärsignal übertragen, so dass die elektromagnetische Abstrahlung reduziert werden (Idealfall: sich gegenseitig aufheben)

**Physical Layer** el./opt. Eigenschaften (Pegel, Timing, Frequ.) Codierung, Stecker-/Pinbelegung

Sorgt für ungesicherte Übertragung eines Bit-Stroms zwischen zwei Knoten über ein Physikalisches Medium (z.B. W-Lan Wellen). Für Anpassung an das Übertragungsmedium selbst (geht zu den Driver über)

**Simplex**: Kanal nur in 1 Richtung (TV, Radio), **Halbduplex**: 1 Kanal beidseitig (Funkgerät),

**Vollduplex** (Telefon) – Punkt-Punkt-Knoten miteinander verbunden, Shared Medium (Bus)

Synchron: Takt-Knoten synchronisiert - Asynchron: Abtasten bis Startbit (0), lesen auf Stopbit

**Data Link Layer**

Stellt eine (gegen Fehler gesicherte) Übertragungsstrecke zwischen zwei direkt verbundenen

Knoten zur Verfügung; Fehlererkennung/-korrektur, Frameerkennung, Flow Control,

Addressierung Teilnehmer (MAC-Adresse), Medium-Access-Control (Master-Slave etc)

Frameerkennung Asynchron: Beginn Frame mit Startbit angezeigt.

**Asynchrone** Übertragung: Ruhephase (0 status) von min. 1 Zeichendauer für Sync [nach Bitfehler]

Hat Start und Stopbit (Normalerweise 1) für jedes Zeichen. (8 Datenbits → 10 Gesendete Bits)

**Synchrone** Übertragung: Frames haben Start&Endflag, wenn Bitmuster in Datenblock=Flag, mit Bitstopfen abgeändert (nach 5 Einsen, immer eine 0). Keine Ruhephase nur leere Frames

Nachteile langer Frames: Durchsatz [↑ Übertragungsfehler], Effizienz [↑ Verlust von Frame], Zuverlässigkeit [↑ Unentdeckte Fehler], Jitter [↑ Delay von regelmäßigen Daten]

Error Correction: Backward = Resend | Forward = Try to fix

Hamming Distanz: # unterschiedlicher Bits in einem Code (H=2, 1 Fehler erkennbar | H=3, 1 Fixbar)

Flusssteuerung: Start-Stop-Meldung | Stop-And-Wait-Protokoll [Quittierung]

Deterministischer Zugriff: **Master/Slave**-Verfahren 1 Master (Single point of Failure)/n-Slaves (Einfach

Konfiguriert) Langsam zwischen Slaves da über Master, **Token Passing**+Peer-to-Peer/-Management aufwändig

-Tokenverlust, **Frame Passing** wie Token Pass. **Zeitgesteuert** Planung, bei Änderungen neue Planung notwendig

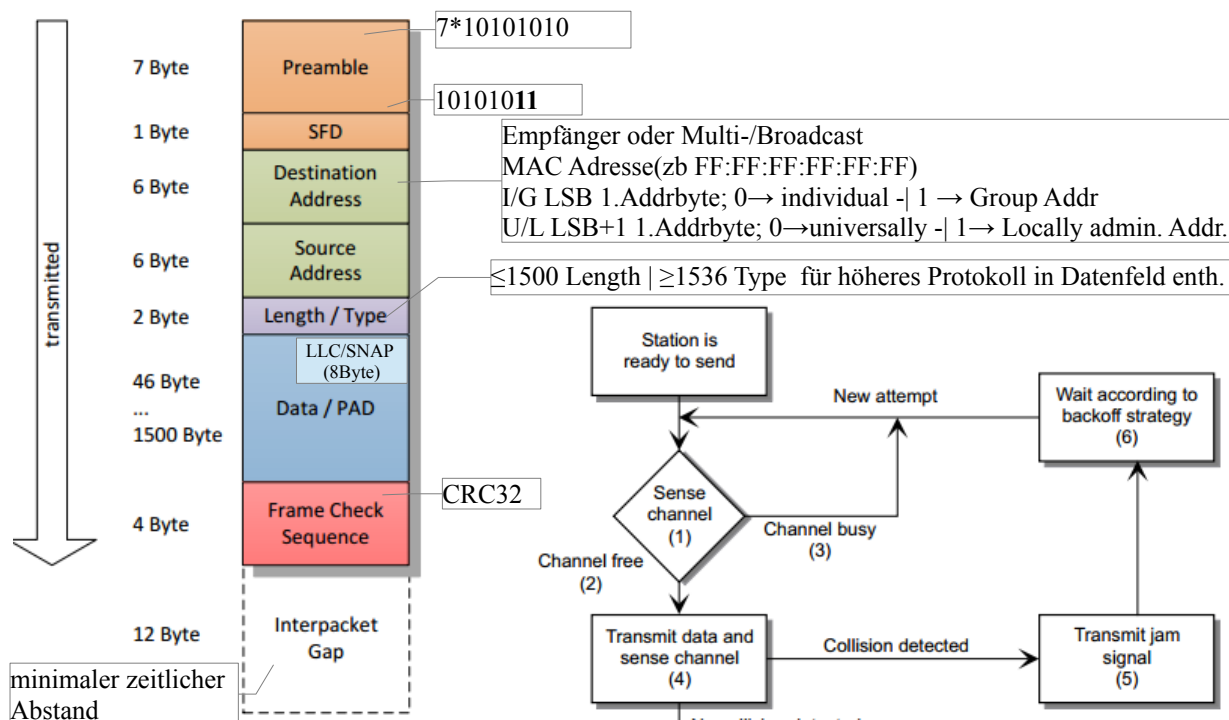
**Ethernet (Local Network)**

Topologie: Bus-, Linien-, [doppel]Ring-, Stern-, Baumtopologie

Unicast [1 Ziel]/ Broadcast [ALLE]/ Multicast [bestimmte Gruppe]

**Ethernet**, Tokenbus **sendet LSB erst**

Tokenring schickt MSB erst



Types in HEX: 0800 = IPv4  
0806 = ARP  
86DD = IPv6

### Switched LAN

Switch Forwarding Database über Sourceadresse befüllt. Idle Entries nach 600s Zeit gelöscht.

Frame Flooding: Sende auf allen Ports [Broadcast] / Broadcast-Domain: Nicht über Router

Mikrosegmentiertes LAN: Nur P2P Verbindungen (kein Repeater[Hub]) ohne Kollisionen

VLAN Bildet eigene Broadcast Domain (Reduzierung Traffic) Aufteilung über VLAN-Tag

VLAN-Tag bestenfalls von Switch hinzugefügt und bei Zielswitch entfernt. Wenn Priorisierung durch Applikation, Endknoten müssen für VLAN konfiguriert werden, da nun VLAN-Tag von Clients hinzugefügt werden.

Switch Leistungsmerkmale:

#Zieladressen, Größe Buffer,

BitRate, ForwardingRate,

Latency, Switching

Mechanismus (Store-and-

Forward [Fehler Filternd,

langsam] / OnTheFly [Fehler

ignorierend, schnell])

Management (Plug&Play)

**Autonegotiation:** Verhandlung

über Geschwindigkeit &

Art (Halb-/Voll Duplex) über

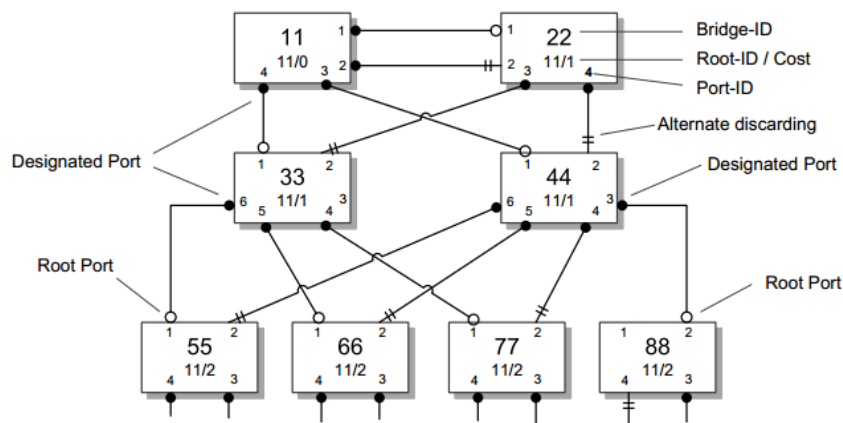
Link Pulses (NLP für 10Mb/s | FLP für >10Mb/s, FLP 17 clock & 16 Bitpulse (für

Konfigurationssettings) abwechselungsweise)

**Scrambler** um 1/0 besser Verteilen mittels 11 Bit Randoms in Linear Feedback Shift Register

Extra Carrier Extension: Erhöhung minimal-Frame größe um Late Collisions zu vermeiden

**Frame Bursting:** Mehrere kleine Frames in größerem Frame zusammen gefügt.



**Network Layer** Datenaustausch zwischen Knoten vereinheitlicht, unabhängig von Ethernet/WLAN, Netz mit mehreren Wegen. Gebraucht um geeigneten Weg zu wählen (Routing)

Flacher Adressraum MAC-Adressen (Keine Aussage über Position im Netz)

Hierarchischer Adressraum IP-Adressen (Positionierung im Netz & Subnetz)

**Router** (Unpackt bis Network Layer)

- +verwendet optimaler Pfad +Komplette Trennung von Netzen +Barriere für Broadcasts
- Teurer als Bridges -Schwierigere Konfiguration -Nonroutable Protocols (ARP)

**Routing Information Protocol** für innerhalb in Subnetz (UDP)(prevents Loops) (Grosse Netze: OSPF)

**Exterior Gateway Protocol** Routing zwischen Subnetzen (TCP)(erreichbarkeit) (**BorderGP**)

**Multi-Home Hosts** mit mehreren Interfaces in mehren Subnetzte angeschlossen

**Classful-Routing**

Ohne Netzmasken, mit 1.Adressbit definiert

**Classless-Routing**

Subnetzmaske definiert

Netz. Tiefste

Addr.=Netz

Höchste

Addr.=Broadcast

Klasse	Adressbereich	Anzahl Netze	Anzahl Knoten
A	1.0.0.0 – 127.255.255.255	128	16'777'214
B	128.0.0.0 – 191.255.255.255	16'384	65'534
C	192.0.0.0 – 223.255.255.255	2'097'152	254
D	224.0.0.0 – 239.255.255.255	Multicast-Adressen	
E	240.0.0.0 – 247.255.255.255	Reserviert für künftige Nutzung	

**Hierarchisches Routing** Routing Tabelle der Knoten= eigenes Subnetz & Default Router

**MaximumTransferUnit:** max Frame Size, überstiegen -> Fragmentierung

**Fragmentierung:** Jedes Fragment erhält eigenen Header, original Header nicht mehr darin

**IHL** Ip Header Double Words(32 Bit)

Minimum für IHL ist 5

**Total Length** in Bytes (8Bit)

**Flags** [0] [0/1 May/Dont Fragment] [0/1 Last/More Fragments]

**Offset** Startpunkt des Fragments (64Bit)

**TTL** Remaining #Hops **Protocol** 1=ICMP | 6=TCP | 17=UDP

1. Byte (Oktett)								2. Byte (Oktett)								3. Byte (Oktett)								4. Byte (Oktett)																																																	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																																										
Version				IHL				Type of Service								Total Length																																																									
Identification Number																Flags		Fragment Offset																																																							
Time to Live								Protocol								IP Header Checksum																																																									
IP Source Address																																																																									
IP Destination Address																																																																									
Optionen																/		Padding																																																							

Geg.:

IP-Adresse von Host: 192.168.37.1

Netz-ID: 192.168.32.0

Vorgehen:

37=> 00100101  
???????? AND

32=> 00100000

Ges.:

Subnetmask

Wir wissen, dass die Subnetmaske in der Bitschreibweise folgendes Muster hat: (111...000)

Also müssen wir ??????? mit 00100101 AND-Verknüpfen, um 00100000 zu erhalten. Durch das Muster kann man herleiten, dass nur bis zur 3. Stelle mit Einsen befüllt werden muss/soll!

Also lautet ??????? = 11100000 = 128+64+32 = 224

=> Subnetmask = 255.255.224.0

**OR** => Analog mit (Subnetmask)-1 für Broadcast-Adresse

**ICMP Protocol:** Teil von IP, obwohl es sich verhält wie höhere Schicht. Wird als Errorhandling-Protocol von IP verwendet – Fehlerursache Melden. Hat ICMP-Header und ICMP-Datenbereich verpackt in IP-Package.

0-Echo Reply/3-Destination Unreachable/4-Source Quench/5-Redirect/8-Echo/11-TimeExceeded/12-ParameterProblem/13-Timestamp/14Timestamp-Reply/15-Information Request/16-InformationReply

**Address Resolution Protocol** Broadcast ARP-Request-Pack (Who-Is?). Antwort in ARP-Cache

**Gratuitous ARP:** Bei Annehmen/Ändern von IP, ARP mit eigener IP -> IP Konflikte erkennen

**ReverseARP** Frage ob jemand seine IP kennt; gegebene MAC Adresse (RARP Server antwortet)

**Internet Control Message Protocol** Für Meldungen wie Dest.Unreachable, Echo, Time Exceed., etc

Ablauf Ping schicken: Überprüfen der IP; Wenn in eigenem Subnetz -> ARP

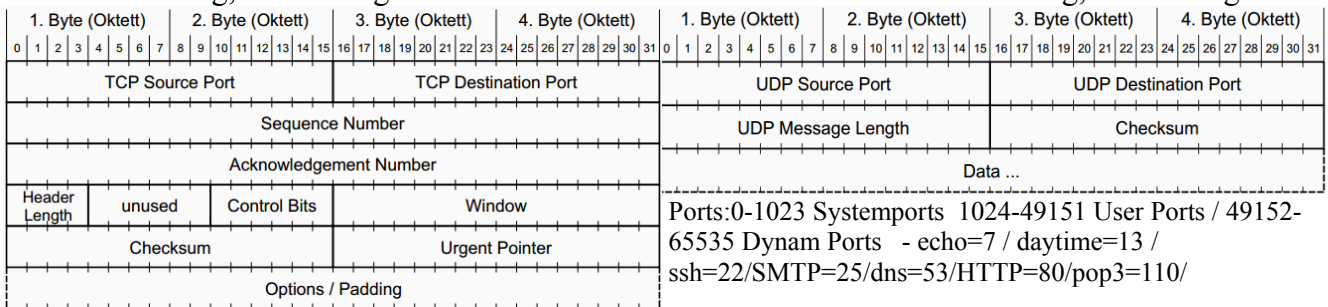
Falls nicht in eigenem Subnetz: Routing Tabelle Konsultieren (Evt. ARP um Router zu finden)

## Transport Layer Aufbau und wieder richtige zusammensetzung Pakete bei Empfänger

Zuverlässig: Intakte Daten, Kein Verlust, Keine Duplikate, Korrekte Reihenfolge

TCP zuverlässig, verbindungsorientiert

UDP unzuverlässig, verbindungslos



TCP - Connection: SYNID=1, SYNID=100 / ACKID=1, ACKID=100, First Package ID=2, ACKID=2 Fenster=1000

Verlust: Gesamtzeit: **Round-Trip Time** /

Neuübertragung nach **Retransmission Time-Out**

Flow-Control: Stop and Wait Package senden wenn ACK vom vorherigen eintrifft / Sliding Window mehrere (z.B. 4) Packages und dann auf ACKS warten

**Überlastüberwachung** (Congestion Control)

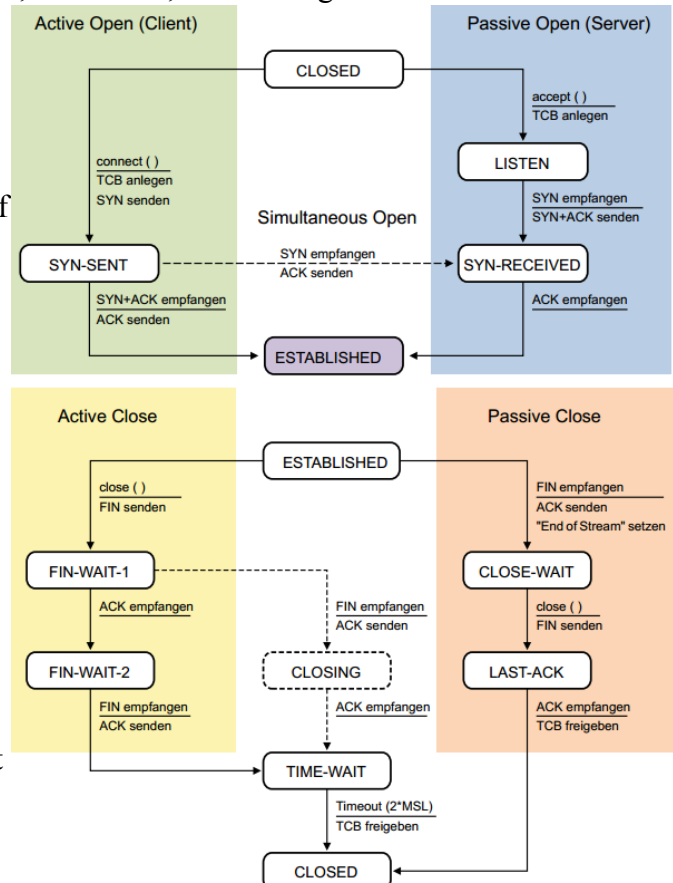
Slow Start; Bei Timeout Schwelle =

Fenstergröße/2, bis schwelle Exponentiell

größere Packages, ab Schwelle Lineares

Wachstum. Repeat bis Max Fenster =

Empfängerfester



## Netzwerk-Applikationen & Protokolle

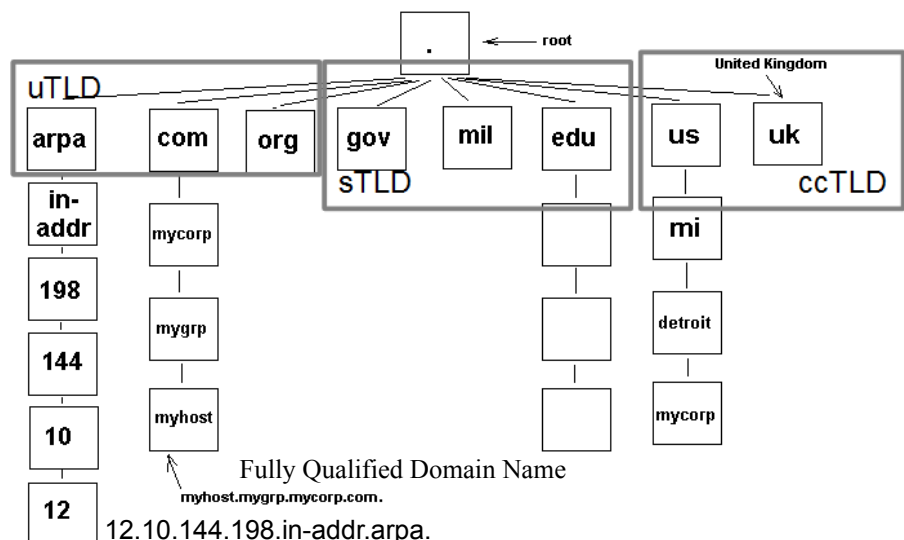
TopLevelDomains:

generische {unsponsored / sponsored} / country code

Gespeicherte IP gewisse TTL, wird mitgeschickt  
Anfrage über Anycast, somit nächster erreicht

BootP = RARP über Router

## Partial DNS Hierarchy



**Session Layer** Logische Verbindung zwischen Sender & Empfänger. Reconnect bei Zusammenbruch (Sockets)  
Internet-Standard Übertragung von Integer Werten; Most Significant bit zuerst. Bei Hardware unterschiedlich, dafür Net to Host funktionen (htons / htonl | ntohs / ntohl)  
In Programm ein Handle auf das Socket (wird von OS verwaltet)

**Presentation Layer** Darstellung von Daten (ASCII, ISO, Unicode, JPEG, **MIME**)

**Quoted Printable** =[ASCII,ASCII] bsp: hü => h=FC

**Base64** Binär/ASCIIencoding mittels Base 64 Tabelle (3 Zeichen zu 4)(MIME),

URL (Uniform Resource Locator) URI (Uniform Resource Identifier)

**Application Layer** Bindeglied zwischen Anwendung und Kommunikationspartner, Protokolle wie FTP, SMTP, HTTP, DNS, DHCP

**DHCP** (Dynamic Host Configuration Protocol) Zentraler IP Vergabe-Server

Ablauf: Broadcast nach DHCP, Offers von DHCP, Auswahl und anfrage, bestätigung -> IP vergeben für lease-time

**SMTP** (Simple Mail Transfer Protocol) Standardprotokoll zum Versenden und Weiterleiten von E-Mail

Nachricht besteht aus 7-Bit-ASCII Zeichen, Mehrere Headerzeilen gefolgt vom Nachrichtentext.

**TFTP** (Trivial File Transfer Protocol) Simples FTP, auf UDP mit ACKs und Timeout für Resend basierent



## Formeln & Zahlen

### Übertragungsmedium

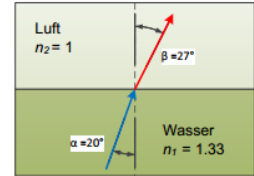
Signaldämpfung [dB] =  $10 \cdot \log_{10} \left( \frac{P_1}{P_2} \right)$  Eingangsleistung P1, Ausgangsleistung P2

Signaldämpfung [dB] =  $20 \cdot \log_{10} \left( \frac{U_1}{U_2} \right)$  Alternativ mittel Spannung U1 (Eingang), U2 (Ausgang)

Dämpfungsbelag typischerweise dB/100m

Optischer Leiter:  $n_1$ : Brechungsindex Kernglas |  $n_2$ : Brechungsindex Mantelglas  
Ein-/Ausfallwinkel  $n_1 \cdot \sin(\alpha) = n_2 \cdot \sin(\beta)$

Totalreflexion: Voraussetzung:  $\alpha$  bis  $\beta \geq 90^\circ$   $\alpha = \arcsin \left( \sin(\beta) \cdot \frac{n_2}{n_1} \right)$



### Physical Layer

Maximale Symbolrate: Doppelte der Bandbreite (2B) in Herz

Maximal erkennbare Zustände:  $Max.[m] \text{ mögliche Zustände} = 1 + \frac{A[\text{maxGrösse Signal}]}{(\Delta V[\text{ungenauigkeit Empfänger}])}$

Informationsgehalt (Bit) eines Symbols:  $[I]_{\text{information}}[S]_{\text{symbol}} = \log_2([M] \text{ mögliche Zustände})$

Maximale Bitrate:  $[R]_{\text{ate}} \leq 2B \cdot \log_2([m] \text{ mögliche Zustände})$

Kanalkapazität:  $C = [B]_{\text{andbreite}} \cdot \log_2 \left( 1 + \frac{[S]_{\text{ignalleistung}}}{[R]_{\text{auschleistung}}} \right)$  zudem:  $M = \sqrt{1 + \frac{S}{R}}$

### Data Link Layer

Effizienz Frames:  $Nettobitrate = \frac{Bruttobitrate \cdot Nutzdaten}{Nutzdaten + Header}$

Durchsatz Erfolgswahrscheinlichkeit  $(1 - p_e[BitErrorRatio])^{N[Frame - Länge]}$   $C_{\text{medium}} = 2 \cdot 10^5 \text{ km/s}$

Optimale Framelänge  $\sqrt{\frac{H[Headerlänge]}{p_e[BitErrorRatio]}}$   $t_{\text{frame}} = \frac{Framesize_{\min}}{Bitrate}$   $t_{\text{transfer}} = \frac{d_{\max}}{C_{\text{medium}}}$

$d_{\max}[istance] < \frac{1}{2} \cdot \frac{Framesize_{\min} \cdot C_{\text{medium}}}{Bitrate}$   $t[ime]_{\text{frame}} < 2 \left( \sum t_{\text{transfer}} + \sum t_{\text{forwarding}} \right)$   
 $t_{\text{frame}} < 2 \cdot t_{\text{transfer}}$

### Transport Layer

Sliding Window Maximaler Durchsatz:  $\frac{WS[Windowsize]}{RTT[Roundtriptime]}$