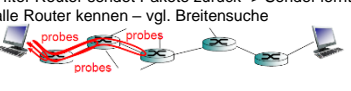


**Traceroute**  
Misst Weg vom Starthost zum Zielhost  
Für alle i:  
- Sende Pakete die nur i laufen können richtung Ziel  
- i.ter Router sendet Pakete zurück -> Sender lernt alle Router kennen – vgl. Breitensuche



**Leitungsvermittlung (Circuit Switching)**  
Benötigte Ressourcen müssen vorab reserviert wdrn Verbindung wird nur zugelassen, falls ausreichend Netzkapazität vorhanden. sonst abgelehnt.  
Dann Senden eines kontinuierlichen Datenstroms. Übertragungsrate garantiert.  
**Paketvermittlung (Packet Switching)**  
Host teilt Nachricht in kleine Pakete auf und schickt sie unabhängig voneinander los.  
Gleichzeitige Pakete müssen sich einen Link teilen und zeitlich hintereinander gesendet werden.  
Router: **Store-and-Forward**.  
Jeder Router muss gesamtes Paket empfangen, bevor er es auf den ausgehenden Link weiterleitet

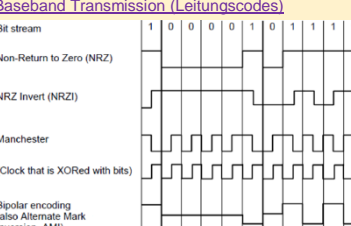
**Schichtenmodell (ISO/OSI)**

**Zusätzlich:**

application
presentation
session
transport
network
link
physical

**Presentation:** Semantik der übertragenen Kommunikation (Kompression, Verschlüsselung, BE, LE)  
**Session:** Sitzungsauf- und abbau Synchronisierung zwischen beteiligten Prozessen

**Baseband Transmission (Leitungscode)**



Bit stream: 1 0 0 0 1 0 1 1 1 1  
Non-Return to Zero (NRZ)  
NRZ Invert (NRZI)  
Manchester  
(Clock that is XORed with bits)  
Bipolar encoding (also Alternate Mark Inversion, AMI)  
Bipolar: abwechselnd + u. – für 1

**Multiplexing (mehrere User 1 Übertragungsmedium)**

**Frequency Division Multiplexing (FDM):**  
- Jeder Benutzer hat eigenen Frequenzbereich

**Time Division Multiplexing (TDM):**  
- Frequenzbereich wird über Zeit geteilt  
- Round-robin  
- Benutzer wechseln sich zeitlich ab

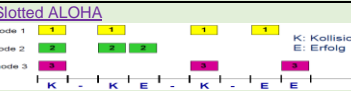
Auch Kombination aus beiden Möglich.

**Cyclic Redundancy Check (CRC): Ethernet und WLAN**

Nur lokal gültig. Zur Identifikation von Nachbarn. Jedes Interface eines Hosts hat eigene MAC-Adresse Bei Ethernet u. WLAN 48 Bit.  
Broadcast-Adresse FF-FF-FF-FF-FF-FF

Jede Netzwerkkarte muss eindeutige MAC-Adresse haben innerhalb eines lokalen Netzwerkes

**Slotted ALOHA**

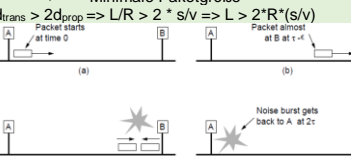


Alle Frames gleiche Größe. Zeit in gleich große Zeitslots unterteilt. Intervall reicht zum Senden des kompletten Pakets. Hosts müssen bzgl Zeit synchronisiert sein.  
Sobald neuer Frame vorhanden. Versuche im nächsten Slot zu senden. Keine Kollision: fertig.  
Kollision: Versuche beim nächsten Slot mit Wahrscheinlichkeit p erneut.  
Vorteile: nur 1 Host: volle Rate, Dezentral, Einfach  
Nachteile: Kollisionen verschwendenden Zeitslots. Sync der Uhren notwendig.

**Mindestlänge von Ethernet Frames**

Worst-Case: Kollision wird erst nach  $2 \cdot d_{prop}$  erkannt. Ziel ist es, dass Sender Kollision noch erkennt, bevor er alle Bits seines Frames losgesendet hat.

Minimale Paketgröße  
 $d_{trans} > 2d_{prop} \Rightarrow L/R > 2 \cdot s/v \Rightarrow L > 2 \cdot R \cdot (s/v)$



Ethernet Switch

Arbeitet auf **Link Layer**.  
Empfang, Zwischenspeicherung und Weiterleitung von Ethernet Frames  
Untersucht MAC Adresse der ankommenden Frames und leitet Frame selektiv nur an richtigen Port weiter. Klassischer Switch hat keine IP-Adresse

Ethernet Switches merken nichts von der Anwesenheit eines Hostes

Selbstlernend  
Muss nicht konfiguriert werden.

**Signalübertragung**  
**Dämpfung:** Längere Leitung -> mehr Dämpfung Leistung/Amplituden verringert  
**Verzerrung:** Frequenzen werden von Übertragungsmedien verschieden stark gedämpft. Meist nur Frequenzen bis zu einem max Wert gut übertragbar  
**Bandbreite:** E-Technik: Frequenzbereich der gut übertragen werden kann  
**Duplex vs Simplex**  
**Vollduplex:** Beide Richtungen gleichzeitig möglich z.B. Kabelübertragung  
**Halbduplex:** Beide Richtungen, aber nicht gleichzeitig z.B. WLAN  
**Simplex:** Nur eine Richtung möglich

**Taktrückgewinnung durch Leitungscode**

Häufige Symbolwechsel nötig, damit Empfänger die Symbole rückgewinnen kann. 1000000 schwierig wie viele 0en.  
**Lösungen:**  
- Synchronie Uhren  
- Manchester Code (Taktfreq =  $2 \cdot \text{Bitfreq}$ )  
- Coderung: z.B. **4B/5B** bildet 4 Bits auf 5 Bits ab mit vielen Wechseln:

Data	Code	Data	Code	Data	Code	Data	Code
0000	11110	0100	01010	1000	10010	1100	11010
0001	01001	0101	01011	1001	10011	1101	11011
0010	10100	0110	01110	1010	10110	1110	11100
0011	10101	0111	01111	1011	10111	1111	11101

Weitere Vorteile: Hohe Baudrate – Effizienz. Gleichspannung unterdrücken(AMI)

**Link Layer**

Frame: Nachricht auf Schicht 2.  
Wird in allen Nodes implementiert (Netzwerkkarte). Nicht in Hubs!

**Übertragung von Frames zw benachbarten Nodes**  
**Rahmenbildung:** Positionsrichtige Erkennung von Zeichen, Erkennung von Blockgrenzen. Frame = Header+Payload. Payload = IP Paket  
**Vielfachzugriff:** Wer darf Medium wann nutzen?  
**Fehlererkennung/-Korrektur:** Umgang mit Bitfehlern auf physical layer. Hinzufügen von Redundanz  
**Zuverlässige Datenübertragung:** Korrektur von Paketverlusten, korrekte Reihenfolge, Vermeidung von Duplikaten. Bei WLAN teilweise, bei Ethernet gar nicht.

**MAC-Adressen (Adresse der Link Layer)**

Nur lokal gültig. Zur Identifikation von Nachbarn. Jedes Interface eines Hosts hat eigene MAC-Adresse Bei Ethernet u. WLAN 48 Bit.  
Broadcast-Adresse FF-FF-FF-FF-FF-FF

Jede Netzwerkkarte muss eindeutige MAC-Adresse haben innerhalb eines lokalen Netzwerkes

**Carrier Sense Multiple Access (CSMA)**

**Carrier Sensing:** Mitlauschen am Kanal.  
Kanal frei: Beginne Übertragung.  
Kanal belegt: Verschiebe Übertragung.

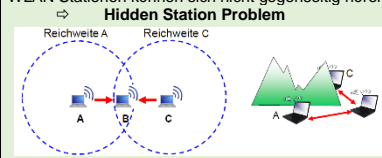
1-persistent: Sende sobald Kanal wieder frei  
p-persistent: Sende im nächsten Slot, mit W'keit p falls Kanal frei ist.  
Non-persistent: Warte Zufällige Zeit und prüfe erneut, ob Kanal frei. => **Ethernet**

Wegen  $d_{prop}$  erkennen Sender erst verspätet, ob es zu Kollisionen kommt.  $d_{prop}$  hat Einfluss auf Kollisionswahrscheinlichkeit.  
Bei spät erkannter Kollision ist losgesendetes Paket wertlos.

**Vielfachzugriff bei WLAN 802.11**

CS: Wie Ethernet wird vor Senden Medium abgehört Collision Detection nicht möglich, weil:  
WLAN ist halbduplex: empf. Signal sehr schwach  
WLAN Stationen können sich nicht gegenseitig hören

**Hidden Station Problem**



Kollisionen müssen beim Empfänger erkannt werden!

**Switch Forwarding**

**Forwardingtabelle** enthält Info, an welchen Port ein Frame weitergeleitet werden muss:  
Ziel MAC-Adr | Ziel Port | TTL  
Selbstlernend: Bei ankommenden Frame werden Infos des Absenders gespeichert.

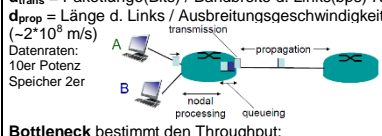
Nachschlagen, ob Eintrag mit Ziel MAC schon in Tabelle.  
**Vorhanden:** Weiterleiten an Zielport. Falls Zielport == Quellport. Frame verwerfen

**Sonst:** Fluten. Weiterleiten an alle Hosts mit Ausnahme des Senders. Auch die, die er schon weiß, was dranhängt.


**Paketverzögerungen/-verlust**  
Verlust: Pakete verworfen, wenn Puffer nicht frei  
Verzögerung: durch Pufferung

$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$   
 $d_{trans} = \text{Paketlänge(Bits)} / \text{Bandbreite d. Links(bps)}$   
 $d_{prop} = \text{Länge d. Links} / \text{Ausbreitungsgeschwindigkeit}$   
( $\sim 2 \cdot 10^8$  m/s)

Datenraten:  
10er Potenz  
Speicher 2er



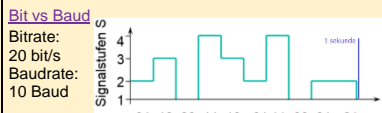
**Bottleneck bestimmt den Throughput:**



**Nyquist (Datenrate D bei unverrauschem Kanal)**  
Bandbreite B; Anz. verw. Signalstufen V  
 $D = 2 \cdot B \cdot \log_2(V)$  [bit/s]  
**Shannon (Datenrate D bei verrauschten Kanal)**  
Gilt zusätzlich zu Nyquist!  
Nutzsignalleistung S; Rauschleistung N  
 $D = B \cdot \log_2(1 + S/N)$  [bit/s]; S/N in dB:  $10 \cdot \log_{10}(S/N)$

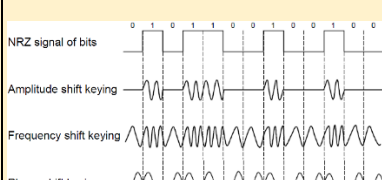
**Bit vs Baud**

Bitrate:  
20 bit/s  
Baudrate:  
10 Baud



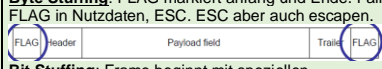
**Passband Transmission**

Nutzsignal ändert Trägersignal  
Bei Frequency Vereinbarung welche Freq 0 und 1



**Rahmenbildung**

Erkennung, wann Frame beginnt und endet.  
**Byte Count:** Zu Beginn jedes Frames Feld, das Anz enthaltener Bytes angibt (Anz. inkl diesem Feld)  
Nachteil: Nach Fehler erneute synchronisation schwer  
**Byte Stuffing:** FLAG markierte anfang und Ende. Falls FLAG in Nutzdaten, ESC. ESC aber auch escapen.

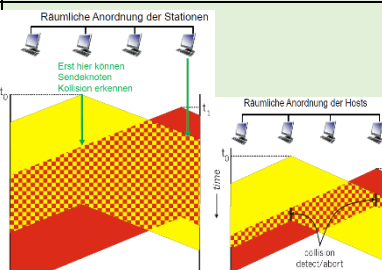


**Bit Stuffing:** Frame beginnt mit speziellen reserviertem Bitmuster. Beim senden wird nach 5 zusammenhängenden 1-er Bits immer ein 0 Bit eingefügt und beim Empfang nach 5 zusammenhäng. 1er Bits immer ein 0 Bit gelöscht.  
Vorteil: Framelänge muss kein vielf. von 8Bit sein.

**Ethernet 802.3 Frames:**

preamble	dest address	source address	type	data (payload)	CRC
----------	--------------	----------------	------	----------------	-----

**Präambel:** 7mal 10101010, dann 1mal 10101011  
=> Synchronisation Sender u. Empfänger  
**Adressen:** je 6 Byte Sender u Empfänger MAC.  
Normalerweise, NW Karte leitet Frame nur an BS weiter, wenn des addresse passt. Ausnahmen: Broadcast oder Promiscuous Mode  
**Type:** 2Byte Art des Netzwerkprotokolls IPv4/IPv6...  
**CRC:** 4Byte  
**Eigenschaften:**  
Verbindungslos: Kein Verb. Aufbau vor Datenaustausch  
keine zuverlässige Verbindung: Frameverlust mögl.  
Vielfachzugriff: Nur bei Punkt-zu-Punkt: Unsl CSMA/CD



**CSMA/CA (Collision Avoidance) bei WLAN**


**Sender:**  
Kanal min. für DIFS frei -> sende kompletten Frame  
Kanal belegt: hier schon exponential Backoff  
=> Unterschied zu CSMA/CD

Höre Kanal ständig ab und dekrementiere Timer, während Zeiten, wo Kanal frei ist. Erneute Übertragung, wenn Kanal frei  
Falls kein ACK eintrifft -> Wieder zu belegt Fall.  
Ggfs backoff Intervall erhöhen.  
**Empfänger:** bestätigt Dateneingang durch ACK nach Zeitspanne SIFS. SIFS < DIFS Priorisierung von ACK

**TODO: GRAFIK**

**Switched Ethernet**

Hub: Alle Leitungen quasi miteinander verbunden.  
Eine einzige Kollisionsdomäne. CSMA/CD notwendig  
Switch: Isoliert jeden Port in eigene Kollisionsdomäne  
Kein CSMA/CD nötig  
Jeder Host direkt mit Switch-Port verbunden.  
Keine Kollision möglich, falls voll-duplex.  
Kein CSMA/CD notwendig.  
Switches speichern Frames zwischen und leiten Frames weiter  
Gleichzeitige Übertragung von A zu A' und B zu B' möglich.



**Schichtenmodell (TCP/IP - Internet)**

Jede Schicht fügt an die Nachricht ihren eigenen Header hinzu

5 Application (HTTP, SMTP, RTP, DNS)  
4 Transport (TCP, UDP)  
3 Network (IP, ICMP)  
2 Link (DSL, SONET, 802.11, Ethernet)  
1 Physical

**Digitale Modulation**

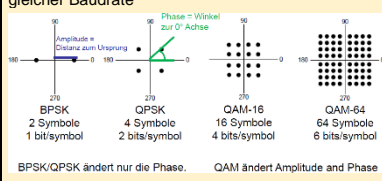
**Modulation:** Umwandlung Bitsequenz in übertragbares Signal.  
**Demodulation:** Rückübersetzung beim Empfänger.

**Baseband** (bei drahtgebundener Übertragung):  
Signal beinhaltet Frequenzen 0 bis fmax und wird direkt in diesem Frequenzbereich übertragen.

**Passband** (bei drahtloser Übertragung):  
Nutzsignal in höheren Frequenzbereich verschieben  
Nutzsignal verändert Trägersignal  
Rückgewinnung am Empfänger durch Demodulation

**Passband: Kombination von Modulationsarten**

ASK und PSK oft kombiniert -> höhere Bitrate bei gleicher Baudrate



BPSK/QPSK ändert nur die Phase. QAM ändert Amplitude und Phase

GrayCode als Bitcodes. Dadurch nur wenige Bitfehler

**Fehlererkennung und -korrektur**

Keine Fehlerkorrektur (zu viel Redundanz)  
Bei Fehlererkennung:  
Ethernet: keine Retransmission  
WLAN: Aktive Wiederanforderung des fehlerhaften Blocks durch Link Layer  
**Checksumme** (IP und TCP Header):  
- Bits in Gruppen von 16 Bit Wörtern  
- Summiere alle 16 Bit Wörter unter Berücksichtigung des Übertrags (Übertr addieren)  
- 1er Komplement des Ergebnisses ist Checksum Empfänger:  
- Addiere übertragene Wörter UND Checksum. Übertrag auch wieder addieren!  
- Wenn Ergebnis nur 1er: Kein Fehler

**Vielfachzugriff**

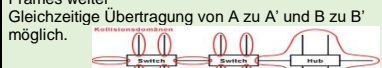
Geteilter Broadcastkanal – Interferenz == Kollision falls mehrere Stationen gleichzeitig senden.  
**Multiple Access Control:** Algorithmus, der entscheidet, wann Host senden darf. Entscheidung muss inband sein (Kein extra Kanal).  
Link hat Kapazität. Wenn mehrere Senden Rate R/M  
**Arten von Multiple Access Control:**  
**Multiplexverfahren:** siehe oben.  
**Random Access Verfahren:** Kollisionen werden zugelassen. Mechanismus um sich von Kollision zu erholen. z.B. Un-/Slotted ALOHA, CSMA/CD/CA  
**Token-Verfahren:** Kollisionen werden verhindert. Nur wer Token hat darf auf Kanal zugreifen

**CSMA/CD (Carrier Sensing + Collision Detection)**

**CD:** Sender (**Netzwerkkarte**) hört während senden Medium weiter ab.  
Sofortiger Abbruch + Jam Signal bei Kollision  
Erneuter Sendeveruch nach zufälliger Wartezeit  
Binary exponential Backoff: mittlere Wartezeit nach jeder erneuten Kollision verdoppelt.  
Sender muss zu **listen while talk** fähig sein.  
LAN: Leicht möglich -> **Vollduplex**  
WLAN: Schwierig. Empfangene Signale viel schwächer als gesendete -> **Halbduplex**

**Switched Ethernet**

Hub: Alle Leitungen quasi miteinander verbunden.  
Eine einzige Kollisionsdomäne. CSMA/CD notwendig  
Switch: Isoliert jeden Port in eigene Kollisionsdomäne  
Kein CSMA/CD nötig  
Jeder Host direkt mit Switch-Port verbunden.  
Keine Kollision möglich, falls voll-duplex.  
Kein CSMA/CD notwendig.  
Switches speichern Frames zwischen und leiten Frames weiter  
Gleichzeitige Übertragung von A zu A' und B zu B' möglich.





<p><b>Network Layer</b></p> <p>Ende zu Ende Verb. Zw. Sender u. Empfänger Router interessieren sich nicht für Schicht 4 / 5</p> <p><b>Adressierung:</b> IP Adressen</p> <p><b>Forwarding:</b> Welches Ausgangsinterface des Routers. Bei Router oft in HW implementiert</p> <p><b>Routing:</b> Berechnung der Wege im Netz. Eintragen d. Ergebnisse in <b>Weiterleitungstabellen</b> (ip Adr. Reichweiten jeweils). Routingprotokolle (konstruieren Routing Tabellen). Meist in SW implementiert IP ist <b>verbindungslos</b>.</p> <p>Link Layer kann unterschiedlich sein (WLAN, Ether..)</p> <p><b>Best Effort:</b> Jeder Router tut sein Bestes, aber keine Garantie bzgl Reihenfolge, Bandbreite...</p> <p><b>IP (20 Bytes Overhead für IP Header)</b></p> <ul style="list-style-type: none"><li>- Adressierungskonventionen TTL: Anz Hops</li><li>- Datagram Format verbleibend</li><li>- Packet handling conventions</li></ul> <table><tr><th colspan="4">32 bits</th></tr><tr><th>Version</th><th>Header length</th><th>Type of service</th><th>Datagram length (bytes)</th></tr><tr><td colspan="2">16-bit Identifier</td><td>Flags</td><td>13-bit Fragmentation offset</td></tr><tr><td>Time-to-live</td><td colspan="2">Upper-layer protocol</td><td>Header checksum</td></tr><tr><td colspan="4">32-bit Source IP address</td></tr><tr><td colspan="4">32-bit Destination IP address</td></tr><tr><td colspan="4">Options (if any)</td></tr><tr><td colspan="4">Data</td></tr></table>	32 bits				Version	Header length	Type of service	Datagram length (bytes)	16-bit Identifier		Flags	13-bit Fragmentation offset	Time-to-live	Upper-layer protocol		Header checksum	32-bit Source IP address				32-bit Destination IP address				Options (if any)				Data				<p><b>Router Architektur</b></p> <p>Queuing, falls Ankunftsrate schneller als Weiterleitung durch Fabric.</p> <p>Bei IP wird Ausgangsport nur anhand der IP Zieladresse bestimmt!</p> <p><b>IP Fragmentierung</b></p> <p>MTU: Verschiedene Link Layer Technologien haben versch. max. Paketgrößen (Ethernet 1500 Byte) Router/Host zerlegt in kleinere Pakete. Zusammenbau am End-Host!</p> <p><b>16-Bit Identifier:</b> Identisch für alle Pakete eines Frames</p> <p><b>Fragmentation Flag:</b> 1: Da kommt noch was, 0: letztes Fragment eines Pakets</p> <p><b>Offset:</b> Byteposition innerhalb des Gesamtpakets, an die das Fragment gehört. (Offset 185: 185 * 8 = 1480, also nach Byte 1480 kommt dieses Fragment)</p>	<p><b>Longest Prefix Matching</b></p> <p>Wenn es für jede IP Adresse einen Eintrag in Forwarding Table gäbe -&gt; zu viel Platzverbrauch. Jeder Port hat einen Adressbereich.</p> <p>Ausgangsport wird so gewählt, dass Ziel-IP mit dem längsten AdressPrefix passt.</p> <p><b>IP Adresse</b></p> <p>32 Bit. Identifiziert Host im Internet, gehört aber logisch eigentlich zum Interface des Hosts. Jedes Interface (am Router mehrere) eigene IP-Adresse.</p> <p><b>Subnetze</b></p> <p>Mehrere Hosts teilen sich gleichen IP Adresspräfix. Innerhalb von Subnetz können sich Hosts ohne Router erreichen.</p> <p>Vorteil: In Routingtabellen müssen nur Subnetzadressen stehen</p> <p><b>Beispiel:</b> 223.1.3.0/24 (die ersten 24 Bits sind für alle Hosts des Subnetzes gleich)</p>	<p><b>Switching Fabric</b></p> <p>3 Typen: Memory, Bus, Crossbar</p> <p><b>Queuing an Eingangsports</b></p> <p>Nötig, falls Fabric langsamer als Ankunftsrate. Head-of-the-Line Blocking: Vorderstes Paket blockiert andere Pakete, obwohl andere Pakete zu einem Ausgangsinterface müssen, das frei ist.</p> <p><b>Queuing an Ausgangsports</b></p> <p>Nötig, falls Ankunftsrate von Fabric die Übertragungsrate des Ausgangslinks übersteigt. Wenn Queue voll -&gt; <b>Paketverluste!</b></p> <p><b>Classful Addressing</b></p> <p>Früher: <b>Feste Länge</b> für Subnetzpräfixe (/8, /16, /24) /24 Netz kann 2<sup>8</sup>(32-24) = 2<sup>8</sup> Hosts haben</p> <p><b>Classless Addressing</b></p> <p><b>Beliebige Länge</b> für Subnetzpräfixe (CIDR) Präfixnotation: 200.23.16.0/24</p> <p><b>Netzmaske:</b> 255.255.255.0 zeigt, welche Bits zum Subnetz gehören</p> <p><b>Spezielle Ipv4 Adressen</b></p> <p>127.0.0.1: Localhost, eigener PC. Netmask 255.0.0.0</p> <p>Private Ipv4 Adressen: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 169.254.0.0/16</p> <p>Beispiel: 192.168.0.0/16 Netzmaske: 255.255.0.0</p> <p>Broadcast: 192.168.255.255 Netzadresse: 192.168.0.0</p>										
32 bits																																													
Version	Header length	Type of service	Datagram length (bytes)																																										
16-bit Identifier		Flags	13-bit Fragmentation offset																																										
Time-to-live	Upper-layer protocol		Header checksum																																										
32-bit Source IP address																																													
32-bit Destination IP address																																													
Options (if any)																																													
Data																																													
<p><b>DHCP (Dynamic Host Configuration Protocol)</b></p> <p>Server weist automatisch IP Adressen zu</p> <p>Eigentlich Schicht 4!</p> <p>Host kann zugewiesene IP Adresse ggfs verlängern.</p> <p><b>Ablauf:</b></p> <ul style="list-style-type: none"><li>- Host sucht DHCP Server: DHCP Discover (optional) Ziel: 255.255.255.255 (Broadcast)</li><li>- DHCP Server antwortet mit DHCP Offer (optional) Ziel: 255.255.255.255 (Broadcast)</li><li>- Host fordert explizit IP Adresse an: DHCP Request</li><li>- DHCP Server weist Adresse zu: DHCP ACK</li></ul>	<p><b>ICMP (Internet Control Message Protocol)</b></p> <p>Error reporting und Router signaling</p> <p>Austausch von Infos zw. Host und Routern</p> <p>ICMP Information wird als IP Paket versendet</p> <table><tr><th>Type</th><th>Code</th><th>description</th></tr><tr><td>0</td><td>0</td><td>echo reply (ping)</td></tr><tr><td>3</td><td>0</td><td>dest. network unreachable</td></tr><tr><td>3</td><td>1</td><td>dest host unreachable</td></tr><tr><td>3</td><td>2</td><td>dest protocol unreachable</td></tr><tr><td>3</td><td>3</td><td>dest port unreachable</td></tr><tr><td>3</td><td>6</td><td>dest network unknown</td></tr><tr><td>3</td><td>7</td><td>dest host unknown</td></tr><tr><td>4</td><td>0</td><td>source quench (congestion control - not used)</td></tr><tr><td>8</td><td>0</td><td>echo request (ping)</td></tr><tr><td>9</td><td>0</td><td>route advertisement</td></tr><tr><td>10</td><td>0</td><td>router discovery</td></tr><tr><td>11</td><td>0</td><td>TTL expired</td></tr><tr><td>12</td><td>0</td><td>bad IP header</td></tr></table> <p><b>Inhalt ICMP Nachricht:</b> Type, Code</p> <p>Erste 8 Bytes des IP Pakets, das Fehler verursacht</p>	Type	Code	description	0	0	echo reply (ping)	3	0	dest. network unreachable	3	1	dest host unreachable	3	2	dest protocol unreachable	3	3	dest port unreachable	3	6	dest network unknown	3	7	dest host unknown	4	0	source quench (congestion control - not used)	8	0	echo request (ping)	9	0	route advertisement	10	0	router discovery	11	0	TTL expired	12	0	bad IP header	<p><b>ARP (Address Resolution Protocol)</b></p> <p>Herausfinden, welche Ziel-MAC zum Next-Hop Router/Host gehört. ARP Paket wird in Ethernet Frame verpackt</p> <p>→ Übersetzen von IP in MAC Adressen</p> <p>Arp Tabelle: IP   MAC</p> <p>Routing Tabelle: IP   Port</p> <p>Switch forwarding Tabelle: MAC   Port</p> <p>ARP direkt auf Ethernet -&gt; kein IP Header</p>	<p><b>ARP Ablauf (Sender und Empfänger in einem Netz)</b></p> <p>A möchte Datagram zu B senden</p> <p>B's MAC nicht in A's ARP Tabelle</p> <p>A schickt Broadcast ARP Query Paket das B's IP Adresse enthält (Ziel MAC: FF-FF-FF-FF-FF-FF)</p> <p>Alle Hosts im LAN empfangen diese ARP Query</p> <p>B empfängt ARP Query und informiert A in Antwort über B's MAC Adresse -&gt; Unicast Frame zu MAC A</p> <p>A speichert IP/MAC paar in ARP Tabelle bis veraltet</p> <p>ARP bedarf keiner Konfiguration (plug and play)</p>
Type	Code	description																																											
0	0	echo reply (ping)																																											
3	0	dest. network unreachable																																											
3	1	dest host unreachable																																											
3	2	dest protocol unreachable																																											
3	3	dest port unreachable																																											
3	6	dest network unknown																																											
3	7	dest host unknown																																											
4	0	source quench (congestion control - not used)																																											
8	0	echo request (ping)																																											
9	0	route advertisement																																											
10	0	router discovery																																											
11	0	TTL expired																																											
12	0	bad IP header																																											
<p><b>ARP Ablauf (Sender u. Empfänger in versch. LANs)</b></p>	<p><b>Routing</b></p> <p><b>Link State (zentral):</b> Jeder Router kennt komplette Topologie. Z.B. OSPF</p> <p><b>Distance Vector (dezentral):</b> Jeder Router kennt nur direkten Nachbarn u. Kosten zu diesem. Nachbarn teilen per Routingnachrichten mit welchen Knoten sie mit welchem Gesamtkosten erreichen können. Z.B. Routing Information Protocol (RIP)).</p> <p><b>Statisch:</b> Manuelle Konfiguration von Forwardingtable</p> <p><b>Dynamisch:</b> Periodischer Austausch von Routinformationen. Änderungen autom. erkannt.</p>	<p><b>Links State vs Distance Vector</b></p> <p><b>Routingnachrichten:</b></p> <p>LS: Jeder Router flutet Infos über seine Links im ganzen Netz</p> <p>DV: Jeder Router informiert seine Nachbarn welche ziele er zu welchen Kosten erreichen kann</p> <p><b>Robustheit:</b> Was wenn ein Router bössartig ist?</p> <p>LS: Router kann falsche Linkkosten ankündigen.Fehler begrenzt, da jeder Router seine eigene Tabelle berechnet.</p> <p>DV: Router kann flasche Pfadkosten ankündigen. Fehler pflanzen sich fort, da Tabelle eines Routers Einfluss auf andere Router hat.</p>	<p><b>Hierarchisches Routing</b></p> <p>Router werden in autonome Systeme (AS) gruppiert.</p> <p><b>Intradomain Routing (für Ziele in anderen Asen)</b></p> <p>Wie sind Gateways aus Nachbarnetzen aus lokalem Netz erreichbar?</p> <p>Protokolle: <b>RIP, OSPF, IGRP</b></p> <p><b>Interdomain Routing (für Ziele im gleichen AS)</b></p> <p>Welche externen Ziele sind über welches Transfer-AS / Gateway erreichbar</p> <p>Protokolle: <b>BGP</b></p> <p>Zusätzliche Routing Policies notwendig: Jeder Router kann bestimmen, welche Nachrichten er bevorzugt.</p> <p>Auch wirtschaftliche Aspekte spielen eine Rolle.</p>																																										
<p><b>OSPF (Open Shortest Path First)</b></p> <p>Router fluten <b>Link State</b> Advertisement Nachrichten (enthalten Infos über alle Nachbarrouter) an alle anderen Router im gesamten AS. Dann Routenberechnung über Dijkstra.</p> <p>OSPF Advertisements werden direkt über IP gesendet (kein TCP oder UDP).</p> <p>Router muss lernen „über welches Interface komme ich aus meinem AS raus zum Ziel, welches sich evtl. in einem ganz anderen AS befindet“. Das steht dann in der <b>Routing Tabelle</b> der Router</p>	<p><b>BGP (Border Gateway Protocol)</b></p> <p>Teile Dem Rest der Welt die Existenz eines IP Präfix mit.</p> <p><b>eBGP:</b> Verbindungen zwischen ASen. Teilt anderen ASen Mit, dass man über diesen Router zu einem bestimmten IP Subnetz kommt.</p> <p><b>iBGP:</b> Verbindungen innerhalb eines AS. Teilt anderen Routern innerhalb eines AS mit, dass man über diesen Router zu einem bestimmten Ziel kommt.</p> <p><b>BGP Attribute:</b></p> <ul style="list-style-type: none"><li>- AS-PATH: Liste von ASen, durch die Prefix Advertisement gelaufen ist.</li><li>- NEXT-HOP: Router im eigenen AS, der das AS mit dem nächsten AS auf dem AS Pfad verbindet.</li></ul>	<p><b>BGP</b></p> <p><b>Policy-based Routing:</b> Beispiele:</p> <ul style="list-style-type: none"><li>- Ignoriere Pfade durch AS Y</li><li>- Gib Routinginfo nicht an Nachbarn AS X weiter</li></ul> <p><b>BGP, OSPF:</b> Welches Ausgangsinterface muss Router verwenden, um zum Gateway Router zu kommen, der über iBGP mitgeteilt wurde.</p> <p><b>Beste Route nach Kriterien:</b></p> <p>1: Local Pref, 2: Kürzester AS Pfad, 3: Route mit am schnellsten erreichbarem Next-Hop (Hot Potato)..</p> <p><b>Hot Potato Routing</b></p> <p>AS will, dass Pakete so schnell wie möglich das eigene AS verlassen, also lokales Gateway mit <b>geringsten Intradomain Kosten</b> wählen.</p>	<p><b>IPv6</b></p> <p>40 Byte Header Kein ARP</p> <p>Keine Fragmentierung und Keine Checksumme</p> <p><b>Notation:</b> 128 Bit in 8 Blöcke, je 16 Bit (4Hex Zahlen), mit „:“ getrennt.</p> <ul style="list-style-type: none"><li>- Führende Nullen darf man weglassen</li><li>- EINMAL dürfen ein oder mehr aufeinanderfolgende Blöcke mit 0000 ausgelassen werden und durch „::“ ersetzt werden.</li><li>- Beispiel: 2001:0db8:0:0:0:0:1428:57ab wird zu: 2001:db8::1428:57ab</li></ul> <p>Es gibt keine /80 Subnetze, weil Host-ID immer genau 64 Bit.</p> <p>IPv4 Notation: ::192.31.20.46</p> <p><b>Tunneling:</b>IPv6 Paket in IPv4 Pak(bei legacy Leitung</p>																																										
<p><b>Transport Layer</b></p> <p>Kommunikation zwischen <b>Prozessen</b> auf Sender u. Empfänger Seite.</p> <p>Teil des <b>Betriebssystems</b></p>																																													

# Abkürzungen

## Einführung

DSL: Digital Subscriber Line

ISP: Internet Service Provider

TCP: Transmission Control Protocol (Netzwerkprotokoll, das definiert auf welche Art und Weise Daten zwischen Netzwerkkomponenten ausgetauscht werden sollen)

UDP:

IP: Internet Protocol (Protokoll das die Grundlage des Internets darstellt)

HTTP: Hypertext Transfer Protocol (Protokoll zur Übertragung von Daten in der Anwendungsschicht)

RFC: Request for Comments (legt Internet Standards fest)

VoIP: Voice over IP

DSLAM: Digital Subscriber Line Access Multiplexer (übersetzt hochfrequente Töne in digitale Signale, bevor Daten zum Modem im Heimnetz kommen)

CMTS: Cable Modem Termination System (Wie DSLAM aber für Kabelmodem)

DHCP Server: Dynamic Host Configuration Protocol Server (Verteilt automatisch Adressen an Hosts in einem Netzwerk)

DNS Server: Domain Name System Server (Weist im Internet einer URL die richtigen IP-Adresse zu)

SAP: Service Access Point (Im Schichtenmodell stellt jede niedrigere Schicht der jeweils höheren Schicht einen SAP zur Verfügung. Somit kann die Höhere Schicht die Services der niedrigeren benutzen)

ISO: International Organization for Standardization

OSI: Open Systems Interconnection

## Network Layer

CIDR: Classless Interdomain Routing (Subnetzteil einer Adresse kann beliebige Länge haben)

# Windows/Linux Befehle

Routing Tabelle anzeigen:

- Linux: route
- Windows: route print

MAC-Adresse rausfinden:

- Windows: ipconfig /all -> physische Adresse
- Linux: ifconfig -> ether

Adresszuweisung

- Linux: ifconfig eth0 200.23.16.4 netmask 255.255.255.0 oder ip addr add 200.23.16.4/24 dev eth0 oder persistent: /etc/network/interfaces
- Windows: Systemsteuerung -> Netzwerk- und Freigabecenter -> Adaptereinstellungen

IP Adresse von DHCP Server anfordern

- Linux: dhclient
- Windows (ipconfig /release)

ARP-Tabellen anzeigen

- Linux: arp
- Windows: arp -a

nmap – Portscanner, scannt auf offene Ports im Netzwerk

TCP Verbindungen anzeigen:

- Windows: netstat -p tcp

# TODO

Schichtenmodell Aufgaben