


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

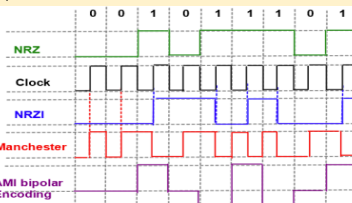


Schichtenmodell (ISO/OSI)
Zusätzlich:

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 (Leitungscodes)
Manchester: XOR Clock mit Bits
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 verschwinden Zeitslots. Sync der Uhren notwendig.



Mindestlänge von Ethernet Frames
Worst-Case: Sender wird erst nach 2 * d_{prop} erkannt.
Ziel ist es, dass Sender Kollision noch erkennt, bevor er alle Bits seines Frames losgesendet hat.
Minimale Paketgröße nötig
d_{trans} > 2 * d_{prop} => L/R > 2 * s/v => L > 2 * R * (s/v)
Diagramm (a) zeigt einen erfolgreichen Frame-Sendevorgang. Diagramm (b) zeigt eine Kollision am Ende des Frames. Diagramm (c) zeigt eine Kollision zu Beginn des Frames. Diagramm (d) zeigt einen Noise Burst, der die Kommunikation unterbricht.

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 Hosts merken nichts von der Anwesenheit eines Switches
Selbstlernend
Muss nicht konfiguriert werden.

Leitungsvermittlung (Circuit Switching)
Benötigte Ressourcen müssen vorab reserviert werden
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

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 übertragbar 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 Leitungscodes
Häufige Symbolwechsel nötig, damit Empfänger die Symbole rückgewinnen kann. 1000000 schwierig wie viele 0en.
Lösungen:
- Synchroner Uhren
- Manchester Code (Taktfreq = 2 * Bitfreq)
- Codierung: 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

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**
Diagramm zeigt zwei Reichweiten A und C, die sich überschneiden, aber B nicht hören können. Ein 3D-Diagramm zeigt die räumliche Anordnung der Hosts und die Zeitachse.

Switch Forwarding
Forwardingstabelle 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_{model} = d_{proc} + d_{queue} + d_{trans} + d_{prop}
d_{trans} = Paketlänge(Bits) / Bandbreite d. Links(bps) R
d_{prop} = Länge d. Links / Ausbreitungsgeschwindigkeit (~2*10⁸ m/s)
Datenraten:
10er Potenz
Speicher 2er
Diagramm zeigt die Komponenten der Verzögerung: transmission, propagation, nodal processing, queueing.

Nyquist (Datenrate D bei unverrauschem Kanal)
Bandbreite **B**; Anz. verw. Signalstufen **V**
D = 2 * B * Id(V) [bit/s]
Shannon (Datenrate D bei verrauschten Kanal)
Gilt zusätzlich zu Nyquist!
Nutzsignalleistung S; Rauschleistung N
D = B * Id(1 + S/N) [bit/s]; S/N in dB: 10 * log₁₀(S/N)
Bit vs Baud
Bitrate:
20 bit/s
Baudrate:
10 Baud
Diagramm zeigt ein Signal mit 10 Symbolen pro Sekunde, wobei jedes Symbol 2 Bits repräsentiert.

Passband Transmission
Nutzsignal ändert Trägersignal
Bei Frequency Vereinbarung welche Freq 0 und 1
NRZ signal of bits
Amplitude shift keying
Frequency shift keying
Phase shift keying

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 markiert anfang und Ende. Falls FLAG in Nutzdaten, ESC. ESC aber auch escapen.
Diagramm zeigt ein Frame mit FLAG, leader, Payload field und Trailer.

Ethernet 802.3 Frames:
preamble dest address source address 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 adresse 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/CD (Carrier Sensing + Collision Detection)
CD: Sender (**Netzwerkkarte**) hört während senden Medium weiter ab.
Sofortiger Abbruch + Jam Signal bei Kollision
Erneuter Sendevorgang 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**

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.
Gfs backoff Intervall erhöhen.
Empfänger: bestätigt Dateneingang durch ACK nach Zeitspanne SIFS. SIFS < DIFS Priorisierung von ACK

CSMA/CA Ablauf:
Diagramm zeigt den Ablauf von DIFS, Data, SIFS, ACK, Backoff, DIFS, Data, SIFS, ACK, Backoff, DIFS, ACK, Backoff rest.

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 f_{max} 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
Diagramm zeigt die Amplitude und Phase zur O'-Achse für BPSK, QPSK, QAM-16 und QAM-64.

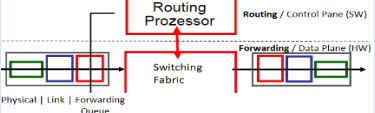
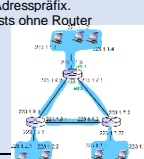
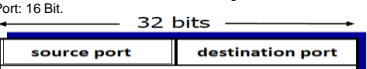
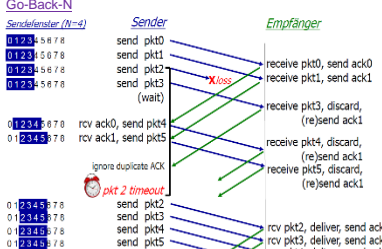
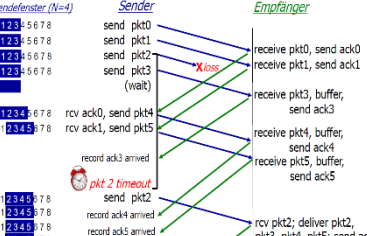
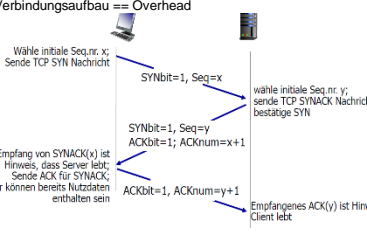
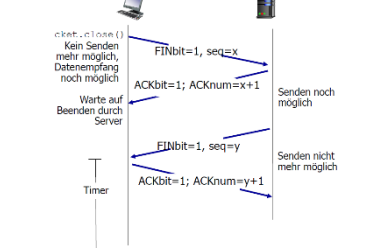
Fehlererkennung und -korrektur
Keine Fehlerkorrektur (zu viel Redundanz)
Bei Fehlererkennung:
Ethernet: keine Retransmission
WLAN: Aktive Wiederanforderung des fehlerhaften Blocks durch Link Layer
Checksumme (IP und TCP Header):
- Bits in Gruppen von 16 Bit Wörtern
- Summiere alle 16 Bit Wörter unter 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 inland 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. Mechanismen 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 Sendevorgang 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.
Diagramm zeigt die räumliche Anordnung der Hosts und die Zeitachse.

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 Hosts merken nichts von der Anwesenheit eines Switches
Selbstlernend
Muss nicht konfiguriert werden.

<p>Network Layer</p> <p>Ende zu Ende Verb. Zw. Sender u. Empfänger</p> <p>Router interessieren sich nicht für Schicht 4 / 5</p> <p>Adressierung: IP Adressen</p> <p>Forwarding: Welches Ausgangsinterface des Routers. Bei Router oft in HW implementiert</p> <p>Routing: Berechnung der Wege im Netz. Eintragen d. Ergebnisse in Weiterleitungstabellen (ip Adr. Reichweiten jeweils). Routingprotokolle (konstruieren Routing Tabellen). Meist in SW implementiert</p> <p>IP ist verbindungslos.</p> <p>Link Layer kann unterschiedlich sein (WLAN, Ether..)</p> <p>Best Effort: Jeder Router tut sein Bestes, aber keine Garantie bzgl Reihenfolge, Bandbreite...</p>	<p>Router Architektur</p>  <p>Queueing, falls Ankunftsrate schneller als Weiterleitung durch Fabric.</p> <p>Bei IP wird Ausgangsport nur anhand der IP Zieladresse bestimmt!</p>	<p>Switching Fabric</p> <p>3 Typen: Memory, Bus, Crossbar</p> <p>Queueing an Eingangsports</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>Queueing an Ausgangsports</p> <p>Nötig, falls Ankunftsrate von Fabric die Übertragungsrate des Ausgangslinks übersteigt. Wenn Queue voll -> Paketverluste!</p>	<p>Classful Addressing</p> <p>Früher: Feste Länge für Subnetzpräfixe (/8, /16, /24) /24 Netz kann 2²⁴(32-24) = 2⁸ Hosts haben</p> <p>Classless Addressing</p> <p>Beliebige Länge für Subnetzpräfixe (CIDR)</p> <p>Präfixnotation: 200.23.16.0/24</p> <p>Netzmasteke: 255.255.255.0</p> <p>zeitg, welche Bits zum Subnetz gehören</p> <p>Spezielle Ipv4 Adressen</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 Netzmasteke: 255.255.0.0</p> <p>Broadcast: 192.168.255.255 Netzmasteke: 192.168.0.0</p>																																											
<p>DHCP (Dynamic Host Configuration Protocol)</p> <p>Server weist automatisch IP Adressen zu</p> <p>Eigentlich Schicht 4!</p> <p>Host kann zugewiesene IP Adresse ggfs verlängern.</p> <p>Ablauf:</p> <ul style="list-style-type: none">- Host sucht DHCP Server: DHCP Discover (optional) Ziel: 255.255.255.255 (Broadcast)- DHCP Server antwortet mit DHCP Offer (optional) Ziel: 255.255.255.255 (Broadcast)- Host fordert explizit IP Adresse an: DHCP Request- DHCP Server weist Adresse zu: DHCP ACK	<p>ICMP (Internet Control Message Protocol)</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>Inhalt ICMP Nachricht: 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>IP Adresse</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>Subnetze</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>Beispiel: 223.1.3.0/24 (die ersten 24 Bits sind für alle Hosts des Subnetzes gleich)</p> 	<p>ARP (Address Resolution Protocol)</p> <p>Herauffinden, welche Ziel-MAC zum Next-Hop Router/Host gehört. ARP Paket wird in Ethernet Frame verpackt -> Übersetzen von IP in MAC Adressen</p> <p>Arp Tabelle: IP Port</p> <p>Routing Tabelle: IP Port</p> <p>Switch forwarding Tabelle: MAC Port</p> <p>ARP direkt auf Ethernet => kein IP Header</p>	<p>ARP Ablauf (Sender und Empfänger in einem Netz)</p> <p>A möchte Datagramm 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 -> 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>ARP Ablauf (Sender u. Empfänger in versch. LANs)</p>	<p>Routing</p> <p>Link State (zentral): Jeder Router kennt komplette Topologie. Z.B. OSPF</p> <p>Distance Vector (dezentral): 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>Statisch: Manuelle Konfiguration von Forwardingtable</p> <p>Dynamisch: Periodischer Austausch von Routinformatioen. Änderungen autom. erkannt.</p>	<p>Links State vs Distance Vector</p> <p>Routingnachrichten:</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>Robustheit: 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 fiasche Pfadkosten ankündigen. Fehler pflanzen sich fort, da Tabelle eines Routers Einfluss auf andere Router hat.</p>	<p>Hierarchisches Routing</p> <p>Router werden in autonome Systeme (AS) gruppiert.</p> <p>Intradomain Routing (für Ziele in anderen Asen)</p> <p>Wie sind Gateways aus Nachbarnetzen aus lokalem Netz erreichbar?</p> <p>Protokolle: RIP, OSPF, IGRP</p> <p>Interdomain Routing (für Ziele im gleichen AS)</p> <p>Welche externen Ziele sind über welches Transfer-AS / Gateway erreichbar</p> <p>Protokolle: BGP</p> <p>Zusätzliche Routing Policies notwendig: Jeder Router kann bestimmen, welche Nachrichten er bevorzugt.</p> <p>Auch wirtschaftliche Aspekte spielen eine Rolle.</p>																																											
<p>OSPF (Open Shortest Path First)</p> <p>Router fluten Link State 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 Routing Tabelle der Router</p>	<p>BGP (Border Gateway Protocol)</p> <p>Teile Dem Rest der Welt die Existenz eines IP Präfix mit.</p> <p>eBGP: Verbindungen zwischen ASen. Teilt anderen ASen Mit, dass man über diesen Router zu einem bestimmten IP Subnetz kommt.</p> <p>iBGP: Verbindungen innerhalb eines AS. Teilt anderen Routern innerhalb eines AS mit, dass man über diesen Router zu einem bestimmten Ziel kommt.</p> <p>BGP Attribute:</p> <ul style="list-style-type: none">- AS-PATH: Liste von ASen, durch die Prefix Advertisement gelaufen ist.- NEXT-HOP: Router im eigenen AS, der das AS mit dem nächsten AS auf dem AS Pfad verbindet.	<p>BGP</p> <p>Policy-based Routing: Beispiele:</p> <ul style="list-style-type: none">- Ignoriere Pfade durch AS Y- Gib Routinginfo nicht an Nachbarn AS X weiter <p>BGP, OSPF: Welches Ausgangsinterface muss Router verwenden, um zum Gateway Router zu kommen, der über iBGP mitgeteilt wurde.</p> <p>Beste Route nach Kriterien:</p> <p>1: Local Pref, 2: Kürzester AS Pfad, 3: Route mit am schnellsten erreichbarem Next-Hop (Hot Potato)..</p> <p>Hot Potato Routing</p> <p>AS will, dass Pakete so schnell wie möglich das eigene AS verlassen, also lokales Gateway mit geringsten Intradomain Kosten wählen.</p>	<p>IPv6</p> <p>40 Byte Header</p> <p>Kein ARP</p> <p>Keine Fragmentierung und Keine Checksumme</p> <p>Notation: 128 Bit in 8 Blöcke, je 16 Bit (4Hex Zahlen), mit „:“ getrennt.</p> <ul style="list-style-type: none">- Führende Nullen darf man weglassen- EINMAL dürfen ein oder mehr aufeinanderfolgende Blöcke mit 0000 ausgeschrieben werden und durch „:“ ersetzt werden.- Beispiel: 2001:0db8:0:0:0:0:1428:57ab wird zu: 2001:db8::1428:57ab <p>Es gibt keine /80 Subnetze, weil Host-ID immer genau 64 Bit.</p> <p>IPv4 Notation: ::192.31.20.46</p> <p>Tunneling: IPv6 Paket in IPv4 Pak(bei legacy Leitung</p>																																											
<p>Transport Layer</p> <p>Kommunikation zwischen Prozessen auf Sender u. Empfänger Seite. (Betrifft Hosts, nicht Router!)</p> <p>Teil des Betriebssystems</p> <p>Transport Layer Multiplexing</p> <p>IP Pakete werden beim Prozess des BS zugeordnet. Port: 16 Bit.</p>  <p>UDP</p> <p>Verbindungslos: Socket definiert durch: Dst IP/Port</p> <p>UDP und TCP, garantieren nicht Delay/Bandbreite!</p>	<p>TCP (alles in SW implementiert)</p> <p>Verbindungsorientiert: Socket definiert durch: Src IP/Port, Dst IP/PORT</p> <p>Zuverlässig: Keine Bifehler, kein Datenverlust, korrekte Reihenfolge</p> <p>Problem: Zuverlässige Übertragung über unzuverlässigen Kanal (IP)</p> <p>ACK: Um zu sagen, ob die Daten angekommen sind, und OK (Checksum) sind.</p> <p>Sequenznummern: Um zu sehen, ob ACK Corrupt ist und um zu sehen, ob das letzte Paket richtig angekommen ist (Ersatz für NACK).</p> <p>Timeout: Schützt vor Daten/ACKs die auf dem Weg verloren gehen.</p> <p>Pipelining: Für verbesserte Performance.</p>	<p>Go-Back-N</p> <p>Empfänger bestätigt immer nur mit Sequenznummer für die gilt, dass alle kleineren bereits empfangen.</p> <p>Bei Timeout werden alle Pakete nochmal geschickt, die noch nicht bestätigt wurden.</p> <p>Selective Repeat</p> <p>Retransmission nur für verlorengegangene Pakete.</p> <p>Empfänger schickt ACKs für jedes einzelne Paket individuell</p>	<p>Go-Back-N</p> 																																											
<p>Selective Repeat</p> <p>Senderfenster (N=4)</p> 	<p>TCP</p> <p>Zuverlässig: Bestätigt einzelne Bytes, nicht Pakete!</p> <p>Pipelining: Misch. aus Go-Back-N u. Selective Rep.</p> <p>Vollduplex: Max Segment Size (MSS) richtet sich nach MTU der Link layer</p> <p>Verbindungsorientiert: Verbindungsaufbau vorher. Sender u. Empfänger initialisieren State Machine</p> <p>Flow Control u. Congestion Control</p> <p>Sequenznummer: Nummer des BYTES, NICHT PACKETNUMMER!</p> <p>Acknowledgement: Sequenznummer des nächsten erwarteten Bytes == ältestes noch nicht empf Byte.</p> <p>Bidirektional: Sequenznummer in eine Richtung sind ACK-Nummer in die andere Richtung</p>	<p>Hybrid von Go-Back-N und Selective Repeat</p> <p>Von Go-Back-N:</p> <ul style="list-style-type: none">- Kumulative ACKs- Nur 1 Retransmission Timer. ältest unbestät Segm <p>Von Selective Repeat:</p> <ul style="list-style-type: none">- Empfangspuffer- Bei Timeout wird nur das verlorengegangene Paket erneut gesendet <p>Auslösen von Retransmissions durch:</p> <ul style="list-style-type: none">- Timeouts- Duplikat ACKs (3) => Fast Retransmit: 3x Duplicate ACK: Indiz für Paketverlust <p>Sender erhält mehr als 3x gleiches ACK => Retransmission des ältesten unbestätigten Segment</p>	<p>TCP</p> <p>TODO: BILD</p>																																											
<p>TCP Verbindungsaufbau (3-Way-Handshake)</p> <p>Verbindungsaufbau == Overhead</p> 	<p>TCP Verbindungsbau</p> 	<p>Flow Control (Empfänger zu langsam)</p> <p>Freier Puffer rwnd = RcvBuffer – (LastByteRcvd – LastByteRead)</p> <p>Empfänger teilt Sender rwnd mit</p> <p>Congestion Control (Netzwerk zu langsam)</p> <p>2 Ansätze:</p> <p>Netzwerk-unterstützt: Router geben Rückmeldung an Hosts bei Überlastung.</p> <p>Ende-zu-Ende: Auslastung des Netzwerks wird durch Beobachten der Verzögerungen und Auftreten von Paketverlusten abgeschätzt. (TCP!!)</p> <p>Additive Increase: Nach jeder RTT wird cwnd um 1 MSS erhöht bis Paketverlust erkannt</p> <p>Multiplicative Decrease: Halbiere cwnd nach erkanntem Paketverlust</p> <p>Window Size <= min{cwnd,rwnd}</p>	<p>TCP Reno</p> <p>Slow Start: Verdoppelt cwnd nach jeder RTT bis zu ssthresh</p> <p>Congestion Avoidance: Dann – vergrößere cwnd um 1 nach jeder RTT</p> <p>Bei Paketverlust/Problem: Halbiere cwnd</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

Winwos/Linux Befehle