**Struktur: Bitübertragung 🡪 Sicherung 🡪 Vermittlung 🡪 Transport** (🡪 Sitzung 🡪 Darstellung) **🡪 Anwendung**

Bitübertragung: **physical layer,** physikalische Übertragung (Transceiver wandelt Daten in Energie um) via Übertragungsmedium

Duplex: **Simplex** (kein Rückkanal, Radio), **Halbduplex** (Datenrichtung ändert sich, Walkie-Talkie), **Vollduplex** (bidirektional,

Ethernet)

Datenrate: R = Bitrate, N = Anzahl Bits, t = benötigte Sendezeit; Laufzeit = Verzögerungszeit zw. Senden &

Ankunft (Latenz)

Bitfehlerrate: (Bit Error Rate) wegen Störeinflüsse, Dämpfung Verbesserung: Kanalkodierung

Kanalkodierung: Hinzufügen von Redundanz (Erhöhung **Hamming-Distanz**: h-1 erkennen, *(h-1)/2* korrigieren), Coderate:

Leitungskodierung: **Fourier-Transf.** (Umwandlung zw. Zeit-/Frequenz), Bandbreite = Frequenzbereich, u(t) = y, t = x, oben U1, unten U-1

NRZ (Non-Return-To-Zero): 1 = voll, 0 = leeres Feld RZ (Return-To-Zero): 1 = halbvoll, 0 = leer

Manchester/Biphase: 1 = linke Hälfte, 0 = rechte Hälfte MLT-3(Multi-lvl-Encoding): Änderung nur bei 1

Ternäre Codes: Umwandl. binär in tenär Quaternäre Codes: Umwandl. binär in quaternär

Modulation: Basisbandsignal durch Modulation in höhere Frequenz, Analog (Amplituden- / Frequenzmodulation)

Digital: **Amplitude Shift Keying (ASK)**: Amplitude h = Leitungskod. **Phase** ..(**PSK**): Phasenänderung **Frequenz** ..(**FSK**): Frequenz umtasten

Medien: **Twisted Pair** (Elektrisch, LAN), **Koaxialkabel** (Wellenleiter, Fernseher), **Lichtwellenleiter** (Glasfaser, WAM)

Pakete/Frames: **TDMA**(Time Division Multiple Access = Aufteilung in Pakete), Frame = Sicherungsschicht, Paket = Vermittlungsschicht

OSI-Modell: **Encapsulation** (vers.): Header an Schicht, weiter n. u. **Decapsulation**(empf): Header von Schicht entfernen, n. o.

Fehlererkennung: Fehler durch elektromagnetsiche Störungen (Rauschen, **SNR** = Signalrauschabstand)

**Paritätsbit**: gerade/ungerade Anz. Bit **Checksum**: Summe aller Bits (IP) **CRC** (Cyclic Redundancy Check): Polynomdivision / XOR (Ethernet, USB)

Fehlerbeh.: **FEC** (Vorwärtsfehlerkorrektur): Redundanz in jedem Paket, Prüfzahl mit übertragen;

**ARQ** (Automatic Repeat Request): **stop-and-wait** ARQ: Senden nur wenn ACK empfangen wurde, *t = N : R*

**Sliding Window**: **Go-Back-n** (Sendef. > 1, Empfangsf. = 1, TCP) **Selective Repeat** (SF = EF, TCP SACK, Neusendung nur fehlerhaftes P.)

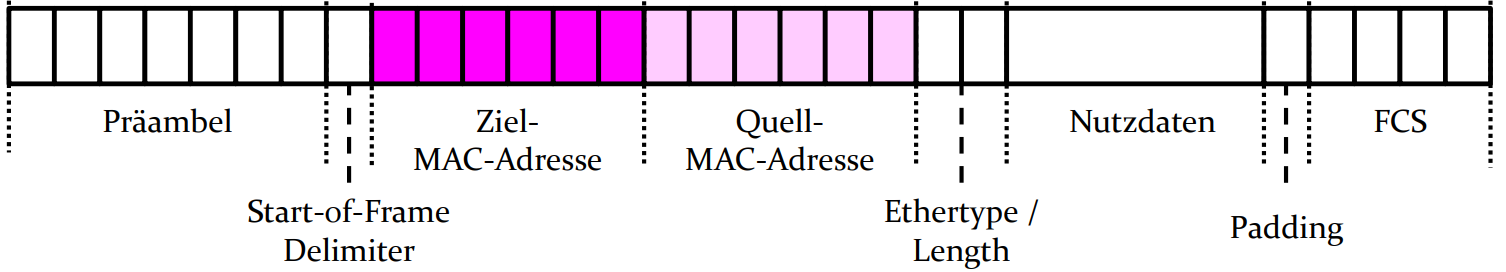
**LAN** (Local Area Network): Adressierung, Kollisionsvermeidung, Topologie (Struktur);

Topologie: **WANs**: vollvermaschtes Netz (k = (n2-n)/2 Verb.), **Bus** (Hub, Kollisionen), **Ring** (IBM Token Ring), **Stern** (Switch)

Half-Duplex: Bus, Kollisionserkennung mit **CSMA/CD**: Medium prüfen, entsprechen Übertragung gleichen Bits?

Full-Duplex: Stern, CSMA/CD integriert, Switch hat **Puffer** (Store and forward, Cut through 🡪 versenden bei Ziel-MAC-Empfang)

VLAN: **Port-based** (VLANs sind Portgebunden), **Tagged** (VLAN-Zuordnung in Header-Fled), Layer3 zum routen zw. VLANs

Ethernet-Frame: Präambel (Uhr-Sync), SFD (Startzeichen),

MAC(Ziel/Quelle), Length (Länge < 1536 = Ethertype), Nutzdaten

(max. 1500), Padding (Füllzeichen für min. Länge), FCS (CRC-32)

WAN Technolog.: **ATM** (LAN/WAN, Switched, für DSL)

**SONET/SDH** (WAN, Multiplexverfahren, Backbone-Infrastruktur für Provider)

Hierarchie:

1. Lokale Netze (LAN, Nutzer), 2. Zugangsnetze (AccessNetworks, WAN, Anschluss an Internet), 3. Kernnetze (Backbone, WAN, ISP)

Autonome Systeme:

ISP-Netze, weltweit eindeutige 32 Bit-Nummern (ASN), Wegewahl über **IGP (Interior Gateway Protocol)**, zwischen AS mit **EGP**

IP: Netzwerk-ID (Präfix), Interface-ID (Suffix), Subnetzmaske (Präfix 1, Suffix 0), Netzwerkadresse (Suffix 0), Broadcast (Suffix 1)

Netzwerkklassen: **CIDR** (Classles Interdomain Routing): beliebige Länge Subnetzmaske, ehemalige Klassen sind Spezialfälle

A /8 (0000-0111), B /16(1000-1011), C /24(1100-1101), D (1110, Multicast), E (1111, reserviert)

Kunde: Präfix von ISP, ISP von zuständigem **RIR** (Regional Internet Registry, EU: RIPE NCC), RIR von **IANA** (Internet Assigne Number Authority)

Private IPv4: **NAT** (Network Address Translation) übersetzt Host-IP in global gültige, Router versteckt NAT hinter seiner eigenen

Spezielle IPv4: 0.0.0.0 (unspezifiziert, wenn Host eigene Adresse nicht kennt), 127.0.0.0/8 (xxx.1 = localhost) Loopback-Adresse

Spezielle IPv6: ::/128 (unspezifiziert), ::1/128(Loopback), Scoped: Link local (automatisch), Unique local (privat), global unicast (normal)

Routingtabelle: Next-Hop (nächster Router), Schnittstelle (Ausgangs-Interface), Metrik (Kostenindex: Hopcount, Kosten, Last, …)

Statisch: Erstellung/Pflegen der Einträge in Routingtabelle von Hand (Kein Overhead, aber keine Fehlertoleranz)

Distance-Vector: Matrix mit bester Metrik speichern 🡪 an Router weiterleiten & lernen, Count-to-Infinity Problem (RIP, IGRP, BGP-4)

max. Hopcount festlegen, Split Horizon (kein Rücksenden an Interface von dem gelernt wurde),… with Poison Reverse

Link-State: Router lernen Topologie (Nachbar lernen & anderen mitteilen), Dijkstras Algorithmus für kürzesten Weg (OSPF, IS-IS)

Autonome Systeme: Innerhalb AS mit **IGP** (OSPF, IS-IS, RIPv2), zwsichen AS mit **EGP** (BGP-4), Routen oft manuell

DHCP: DHCP Discover: Client such DHCP-Server und fordert Informationen an

DHCP Offer: DHCP-Server liefern Informationen an Client

DHCP Request: Client fordert weitere, spezifische Informationen an

DHCP Ack: Antwort an DHCP Request

Eigenschaften IP: paketvermittelnd, verbindungslose, ungesichert, Dienstgüte: Best Effort 🡪 weitere Protokolle benötigt

Header: IPv4: Vers., Head Length, ToS, Total Length, ID, Flags, Fragment Offset, TTL, Type, Checksum, SRC/DST-IP, Options, Padding

IPv6: Vers, Traffic Class, Flow Label, Payload Length, Next Header, Hop Limit, SRC/DST-IP, (Extension)

Fragmentierung: **MTU** (Maximum Transission Unit) 🡪 Fragmente < MTU (durch Router aufgeteilt), Info: Flag & Offset

IP-Multicast: Adressierung Gruppe (IPv4 224.0.0.0-239.255.255.255, IPv6 ff00::/8), **IGMP**(Internet Group Management Protocol)

Source Routing: Loose (muss diese Router passieren, Opt. 3), Strict (darf nur diese Router passieren, Opt. 9)

Route Recording: Route wird als Liste in Optionsfeld gespeichert (Rückverfolgung

Adressauflösung: IPv4: **ARP** (erhält IP, Broadcast in Netz, Host mit IP teilt MAC-Adresse mit, ARP speichert IP & MAC temporär

IPv6: ICMPv6 (wie ARP, verwedet ICMP Messages, zusätzlich: Router-Discovery, Adressduplikate erkennen…)

**ICMP**: Austausch von Fehlermeldungen und Zustandsinformationen

Transportschicht: Lösung für anwendungsabhängige Probleme, Grenze zwischen App & darunterliegenden Schichten

Aufgaben: Bereitstellung zuverlässiger Dienst zur Übertragung, Multiplexing, Programmier-Schnittstelle

Anwendungsaddr.: Multiplexing 🡪 Zuordnung Datenpakete bei gleichzeitiger Nutzung einer Schnittstelle

Transportaddr.: TSAP (Transport Service Access Point): Port (Socket = IP + Portnummer)

Wahl von Port: festgelegt von Server, well-known-Ports (80 für http) 🡪 standartisiert

**TCP**: wichtigstes & am häufigsten verwendetes Transportprotokoll

Eigenschaften: verbindungsorientiert, zuverlässig, geordnet, Fluss-/Staukontrolle, Full-Duplex, Stream-orientiert

Paketaufbau: SRC/DST-Port, Sequence-Number, ACK, Header Length, Flags, Window Size, Checksum, Urgent Pointer, Options

Verbindungsaufbau: Versenden Segment mit SYN-Flag und ACK

Datenübertragung: Versenden mit Seq-Nummer des 1. Bytes, ACK mit erwarteter Seq-Nummer (Seq + Länge + 1) (wenn lückenlos)

Verbindungsabbau: Versenden Segment mit FIN-Flag und ACK

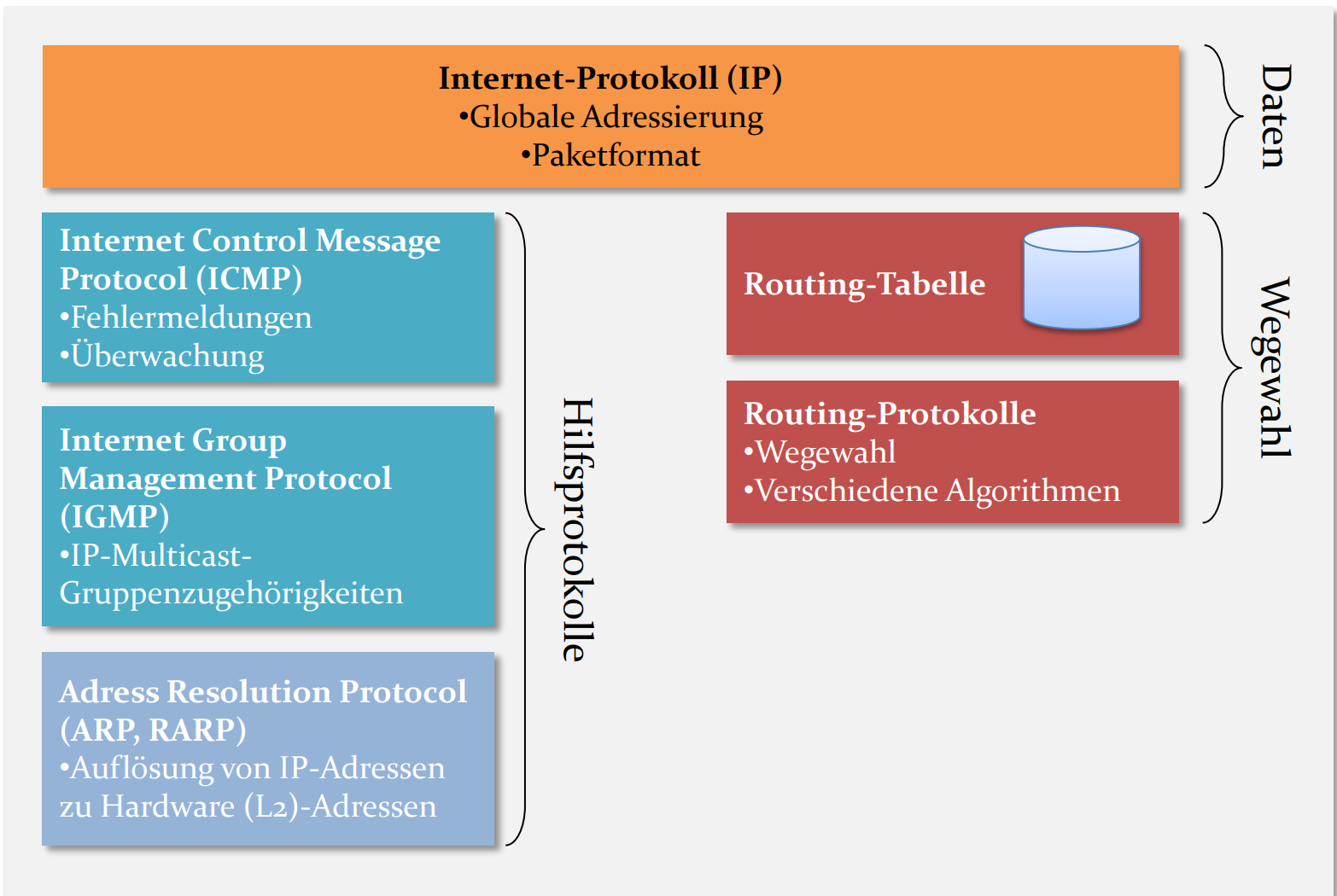
Fast Retransmit: ACK nach jedem Segment, bei Lücke wird gleiches ACK 2x gesendet & Retransmit gestartet

Flusskontrolle: Puffer für Anwendung, Empfänger sendet ACK mit verbleibendem Platz, Sender sendet bei WIN > 0

Staukontrolle: Sendefenster: nur n unbestätigte Bytes gleichzeitig MSS = Maximum Segment Size

Slow Start: Beginn mit n = 1 MSS, jedes ACK: n + 1 MSS, n verdoppeln bis Schwellwert, dann n linear (Congestion

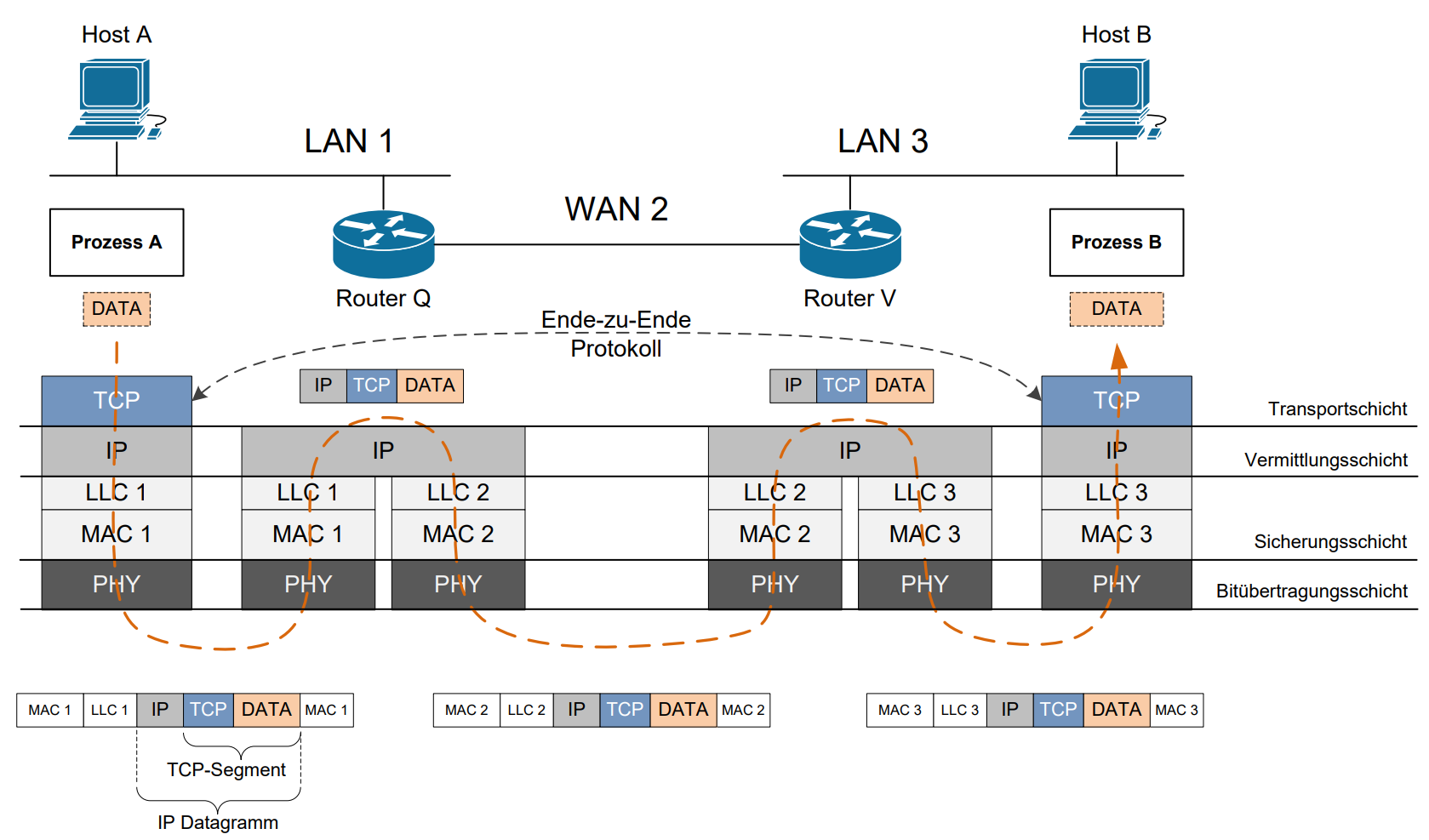
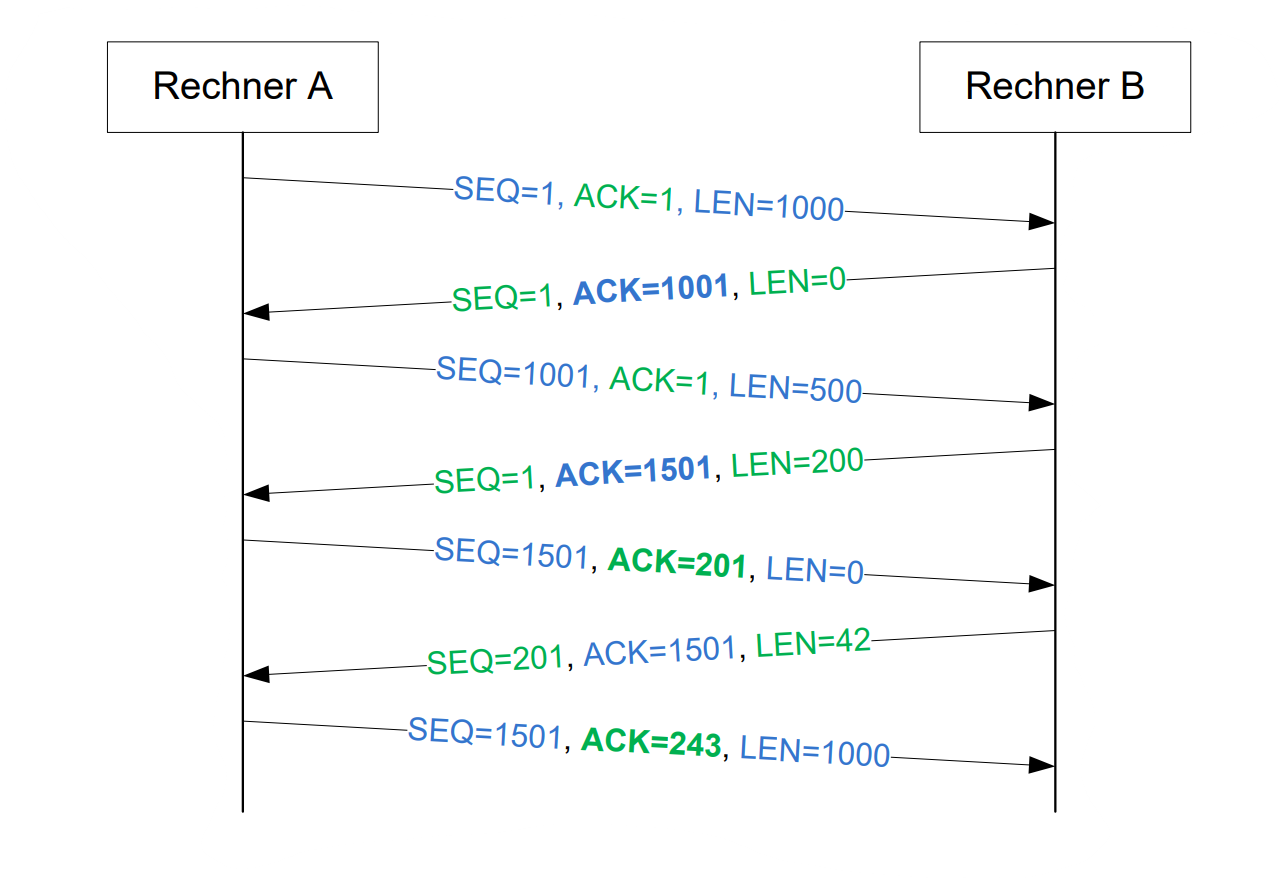
Avoidance), Timeout: n = 1 MSS, CA-Schwelle auf Hälfte der aktuellen Fenstergröße

**UDP**: neben TCP am meisten verwendet (schnell,

unkomplizierter als TCP, aber unzuverlässig 🡪 Telefonie)

Eigenschaften: verbindungslos, unzuverlässig, nachrichtenorientiert, 1-to-many interaction

Header: SRC/DST-Port, Message Length, Checksum(optional)

Socket: bind(): lokale Socketadresse für neuen Socket

connect(): entfernte Scoketadresse (Verbindungsaufbau)

sendto(): Adressierung Datagram an entf. SA

recvfrom(): Absender von Datagramm ermitteln