



Infrastruktur Grundlagen

Einführung in Netzwerke / OSI, Ethernet Protokoll

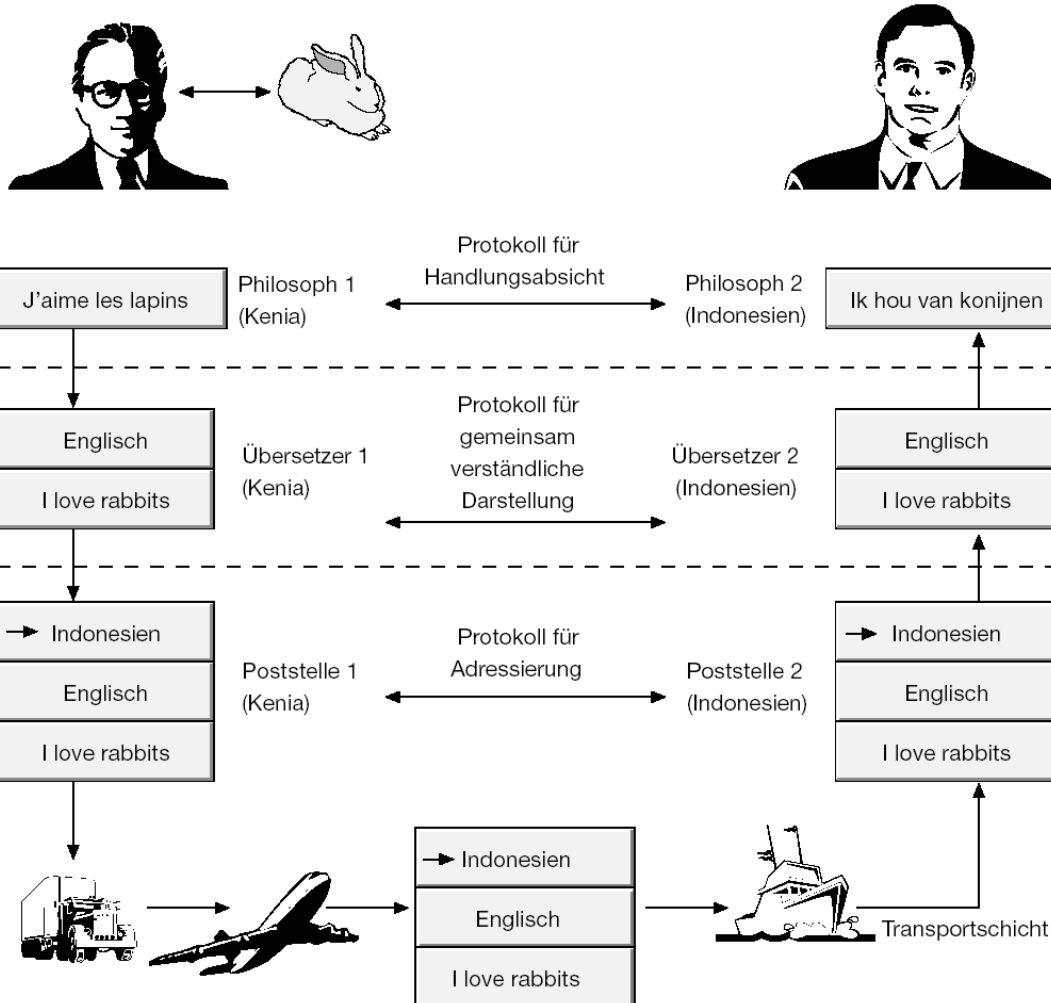
Computernetzwerke / Internet - 1

- **Computernetzwerk:**
 - Zusammenschluss verschiedener Computersysteme wie
 - Server, Laptops, PCs, Drucker, ...
- **Internet:**
 - größte weltweite **technische Computernetzwerk** mit Millionen angeschlossener Systeme
- **Hosts / Endsysteme sind über**
 - **Kommunikationsleitungen** (communication link)
 - und **Paket-Switches** (paket switch)
 - verbunden
- **Übertragung** von PDU („packet data unit“)

Computernetzwerke / Internet - 2

- **Hosts / Endsysteme** in Computernetzwerke nutzen „**Protokolle**“ welche
 - das Format und Reihenfolge des Nachrichtenaustausches regeln
- **TCP/IP Protokoll Suite**
 - **TCP** (Transmission Control Protocol)
 - **IP** (Internet Protocol)

Was ist ein Protokoll?

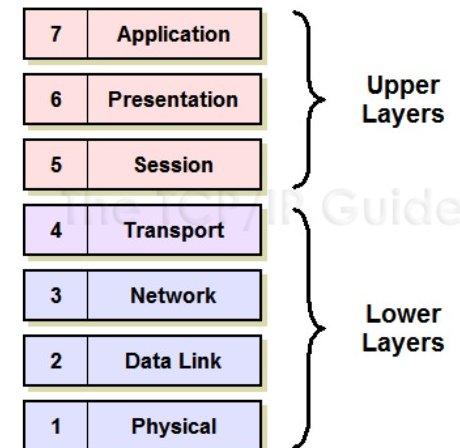


ISO/OSI Referenzmodell

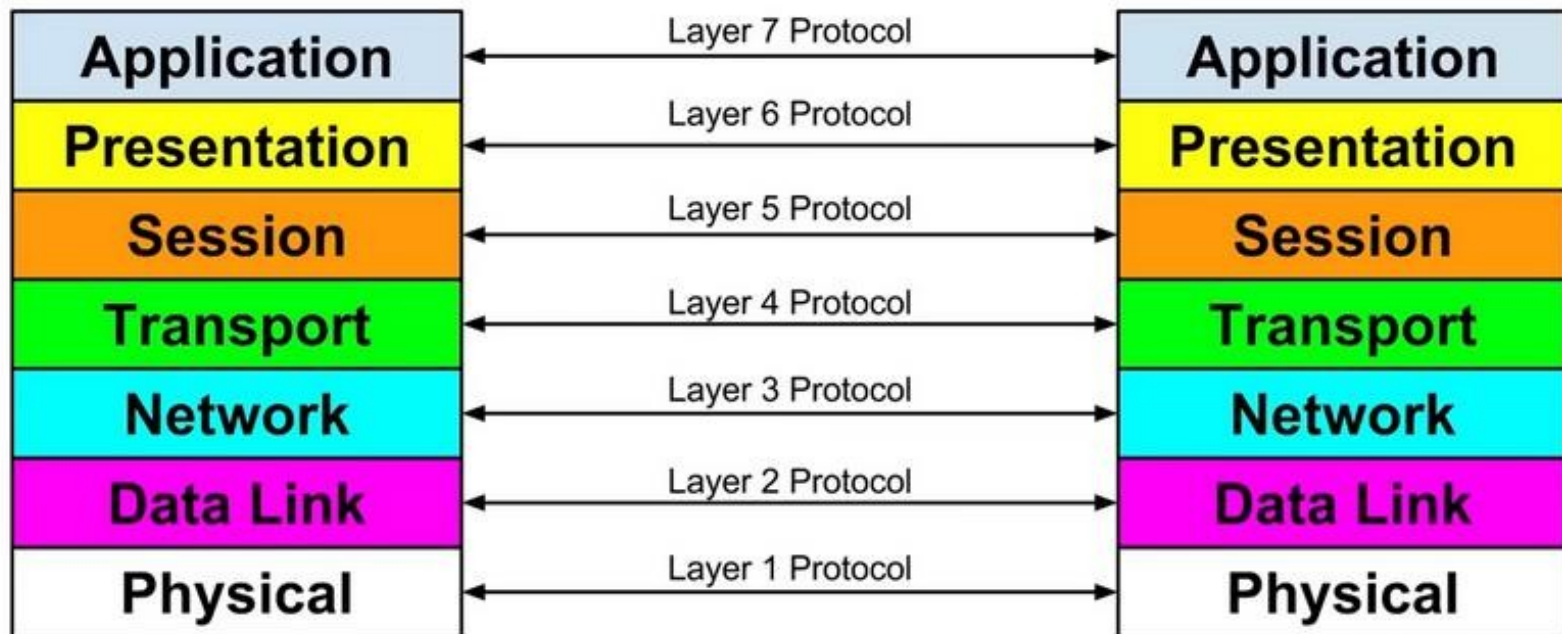
- OSI = **O**pen **S**ystems **I**nterconnection **M**odel
- **Referenzmodell für Netzwerke**
- 1983 von der ISO als Standard veröffentlicht
- **Ziel:**
 - Die Kommunikation unterschiedlicher technischer Systeme zu ermöglichen
- **OSI Modell** ist in **Schichten** - sog. **Layers** - aufgeteilt

ISO/OSI Referenzmodell

- Unterteilt in **7 Schichten**
- Je **niedriger** die Nummer im Stack desto „**Hardware-näher**“
- Je **höher** die Nummer desto „**softwarelastiger**“
- **Upper Layer**
 - Interaktion mit dem User
 - Implementierung von Software die über das Netz kommuniziert
- **Lower Layer**
 - Ziel ist der Transport der Daten

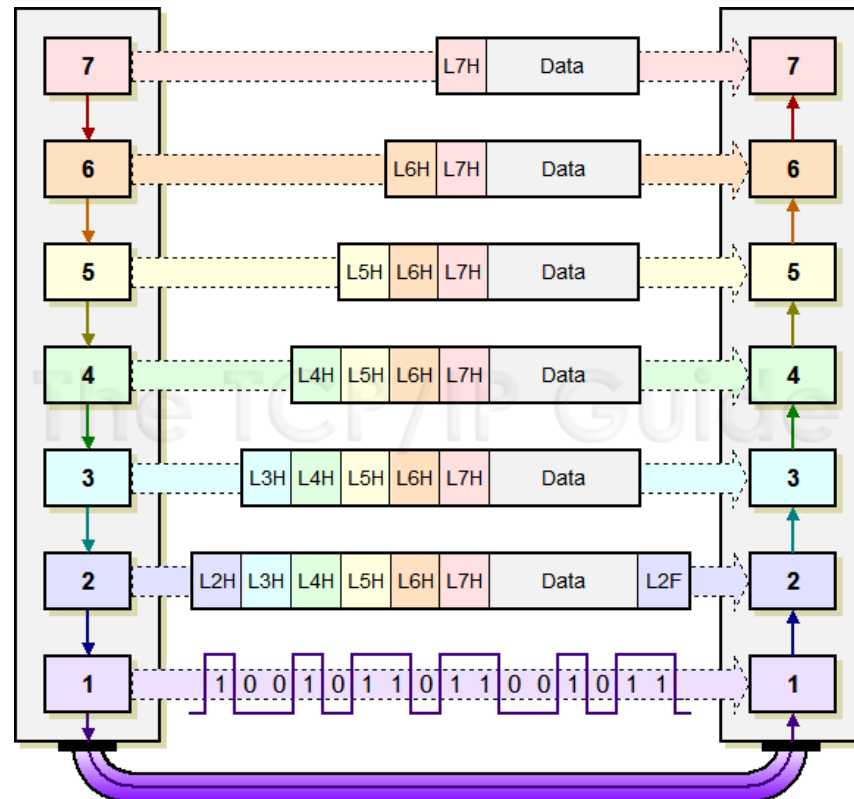


ISO/OSI Referenzmodell

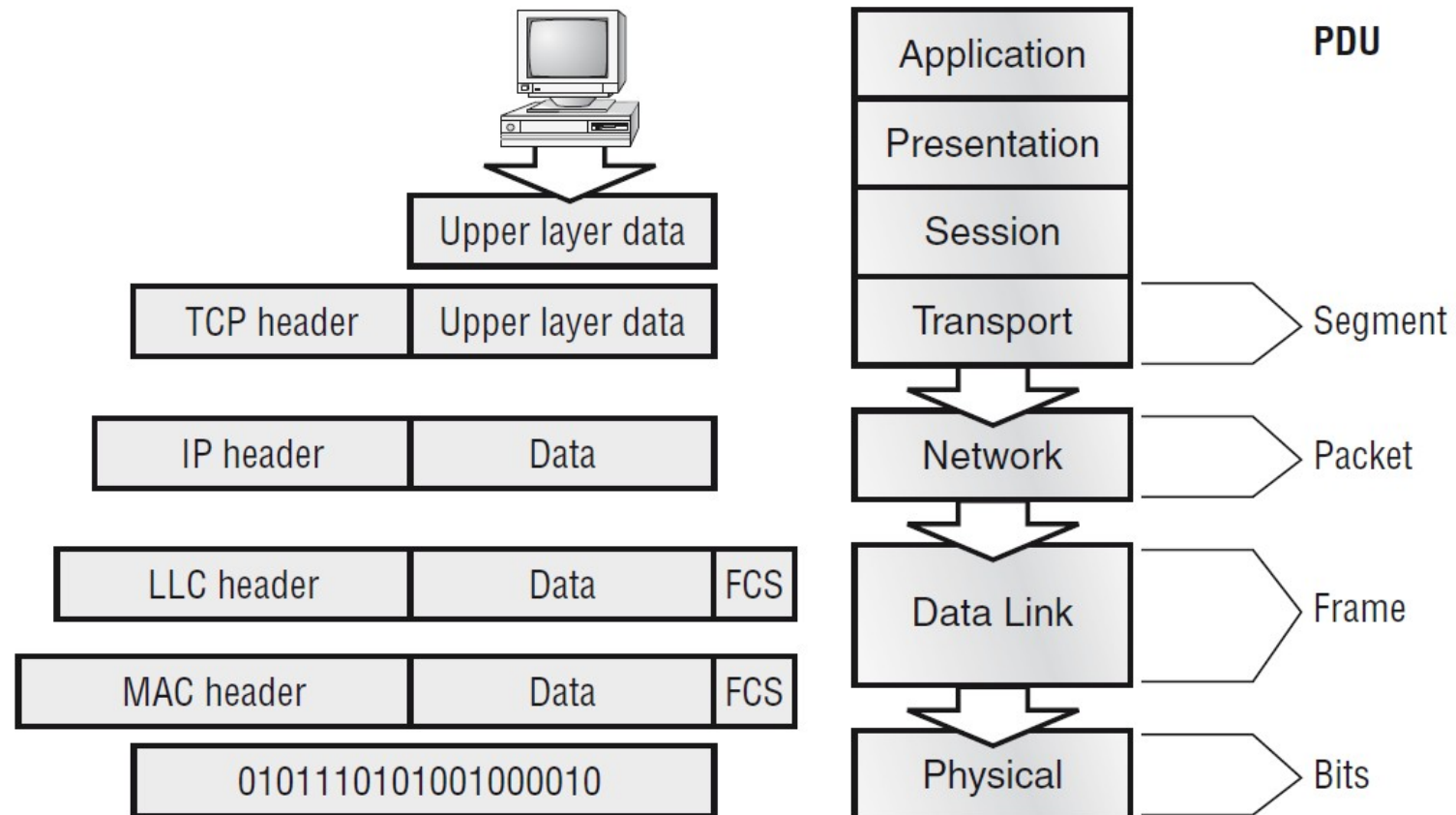


ISO/OSI Referenzmodell - Datenkapselung

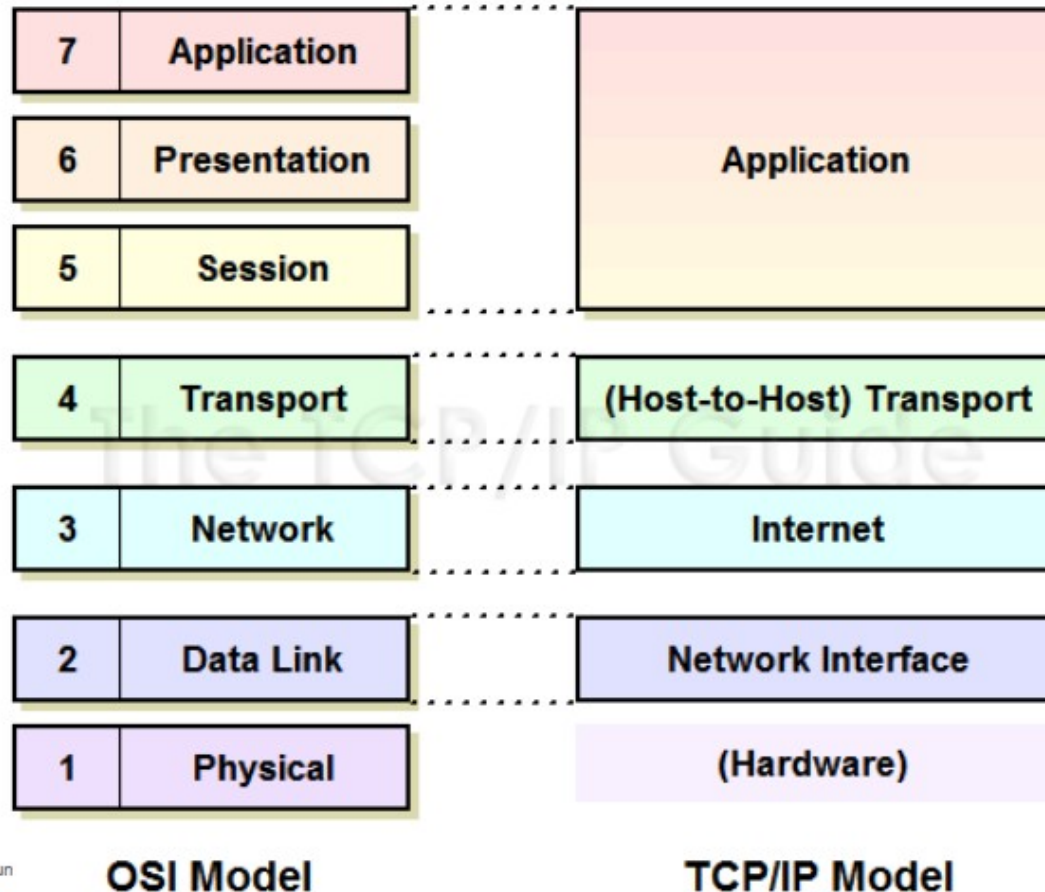
- **Daten** der jeweilig **höheren Schicht** werden in ein **eigenes Paket** gepackt und mit einem eigenen **Layerheader** versehen



Überblick der Datenkapselung



Vergleich ISO/OSI vs. TCP/IP Modell



Protokollschichten – Layers

- **Anwendungsschicht**

- application layer
- Protokolle für Anwendungen z.b. HTTP für Web, IMAP & SMTP für Email

- **Transportschicht**

- **transport layer**
- Überträgt Nachrichten der Anwendungsschicht
- **TCP** (Transmission Control Protocol)
- **UDP** (User Datagram Protocol)

- **Netzwerkschicht**

- **network layer**
- reicht Datagramme von einem Host zum anderen
- Wegfindung (Routing)
- IP Protokoll

Protokollschichten – Layers

- **Sicherungsschicht**

- data link layer
- Übertragung von „frames“
- z.b. über **Ethernet**, WLAN, PPP (Point-to-Point Protocol)

- **Bitübertragungsschicht**

- physical layer
- Transport der Bits je nach Trägermedium (Kupferdrähte, Glasfaser, Satellit, ...)

Trägermedien

- **geführte Medien**

- festes Medium: Lichtwellenleiter, Kupferdraht, Koaxialkabel

- **nicht geführte Medien**

- WLAN, Satellitenkanal

- **Beispiele**

- Twisted-Pair
- Koaxialkabel
- Glasfasern
- Terrestrische Funkverbindungen
- Satellitenfunkverbindung

Verzögerung, Verlust, Datendurchsatz

- in paketvermittelnden Netzwerken
- **Verzögerung:**
 - durch Verarbeitung, Warteschlangen, Übertragung, Ausbreitung
- **Paketverlust:**
 - z.b. durch überschreiten einer Warteschlangenkapazität
- **Datendurchsatz:**
 - Ende-zu-Ende Durchsatz
 - Geschwindigkeit in Bit/s
 - „Flaschenhals“ – Engpassleitung (Bottleneck)

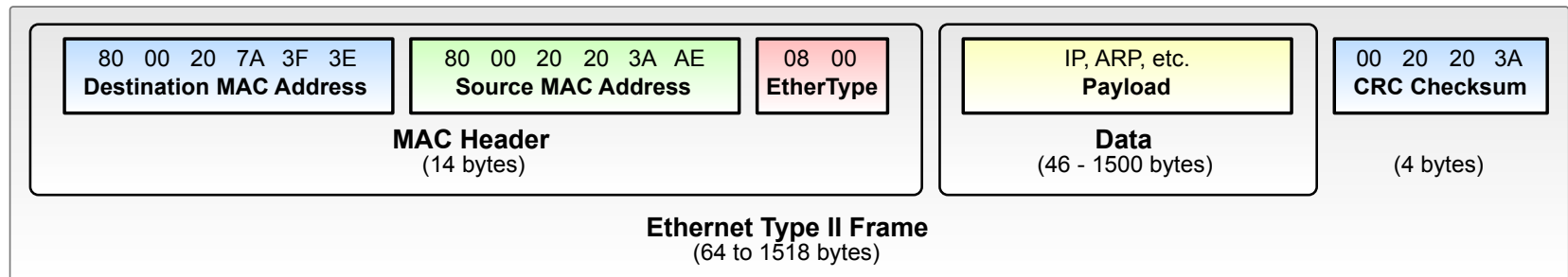
Paket- vs. Leitungsvermittlung

- **Paketvermittlung:** (packet switched)
 - Informationen werden in einzelne Pakete aufgeteilt und über virtuelle Verbindungen übertragen
 - **Vorteil:** Leitung wird „geteilt“, effizienter
 - **Nachteil:** Stau von Paketen möglich, Nicht verzögerte Übertragung kann nicht garantiert werden
- **Leitungsvermittlung:** (circuit switched)
 - Reservierung der Netzwerkressourcen für die Dauer der Kommunikationssitzung
 - z.b. traditionelle Fernsprechnetze
 - **Nachteil:** Ressourcen vollständig belegt (auch wenn ungenutzt)
 - Trend geht zur Paketvermittlung (z.b. VoIP in Telefonie)

Ethernet Protocol – IEEE 802.

2	802.1 Internet-Working	802.2 Logical Link Control (LLC)			
		802.1 Media Access Control (MAC)			
1		802.3 Ethernet	802.4 Token-Bus	802.5 Token-Ring	802.11 Wireless LAN

Ethernet Frame



Netzwerktopologien

- **Physische Anordnung von Netzwerkstationen**
- Verbunden über **Kabel oder Funknetz**
- Bestimmt den Einsatz von Hardware & Zugriffsmethoden
- In weiterer Folge Einfluss auf:
 - Das Übertragungsmedium (Kabel, Lichtwellenleiter, Luft,...)
 - Die Übertragungsgeschwindigkeit
 - Den Datendurchsatz

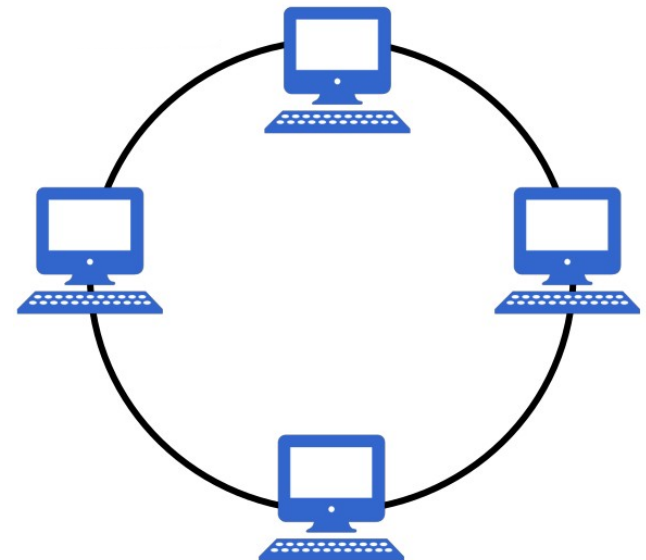
Bus-Topologie

- Eine **gemeinsame Übertragungsleitung**
- Stationen sind **hintereinander geschaltet**
- Um Störungen zu vermeiden werden an den Enden der Leitung sog. **Abschlusswiderstände** eingesetzt
- Trennung des Kabels führt zum Ausfall des Netzwerkes
- **Keine zentrale Netzwerkkomponenten** die den Datenverkehr regelt
- Daten erreichen alle Stationen, Stationen die nicht adressiert sind ignorieren Daten
- Adressierte Station liest Daten und sendet Empfangsbestätigung an den Sender
- **Bei Kollision** von Datenpaketen entsteht eine **Störung** am Bus (= > 2 Stationen senden gleichzeitig)
 - → Nach Zufallszeit versuchen die Stationen erneut zu senden, bis die Daten am Ziel ankommen



Ring-Topologie

- Geschlossene Kabelstrecke
- Netzwerkteilnehmer sind **im Kreis angeordnet**
- An jeder Station kommt ein Kabel an und ein Kabel geht ab
- Trennung des Kabels führt zu Ausfall des Netzwerkes
 - → Ausnahme: Netzwerk kennt Busbetrieb und stellt auf diesen um

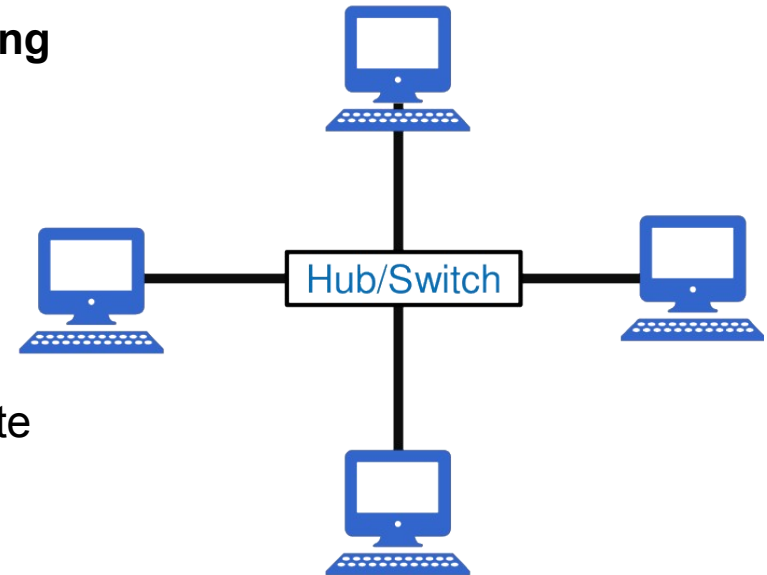


+ Verteilte Steuerung
+ Große Ausdehnung

- Aufwendige Fehlersuche
- Bei Störung = Netzausfall
- Verkabelungsaufwand

Stern-Topologie

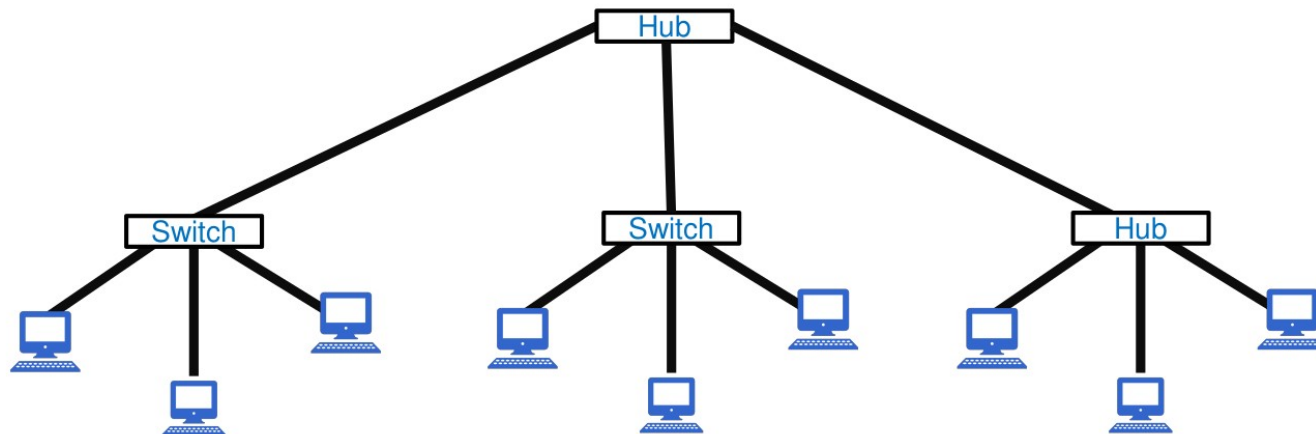
- Eine **zentrale Station** unterhält eine **Verbindung** zu **allen** Stationen
- Jede Station ist somit über eine eigene Leitung angebunden
- **Zentrales** Element ist zumeist ein **Hub** oder **Switch**
- Übernimmt die **Verteilfunktion** der Datenpakete
- Datenlast von Hub/Switch sehr hoch
 - → Alle Datenpakete müssen darüber
- **Ausfall Hub/Switch = Stillstand des Netzwerkes**
- Kann leicht erweitert werden
- Zumeist Kombination aus Bus + Stern üblich



- + Einfache Vernetzung
- + Einfache Erweiterung
- + Hohe Ausfallsicherheit
- Hoher Verkabelungsaufwand
- Netzausfall bei Ausfall des Hubs/Switch

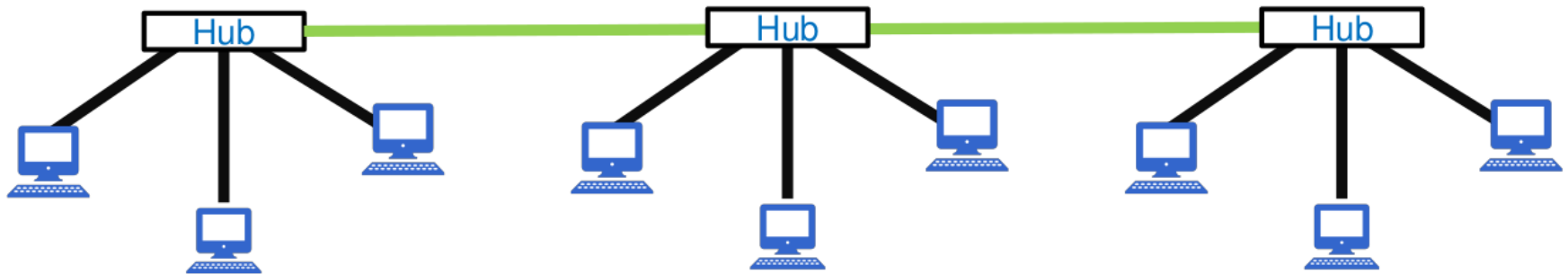
Baum-Topologie

- **Erweiterte Sterntopologie**
- Größere Netze nehmen solche Strukturen an
- Zumeist ein **übergeordnetes Netzwerkelement** das die **Wurzel** des **Baumes** bildet
- Wird wie ein Baum verzweigt



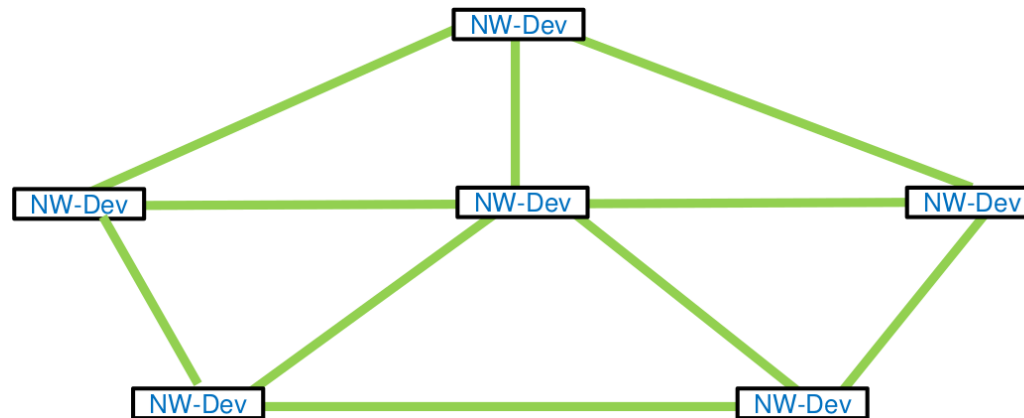
Kombinationen von Topologien

- **Kombination** aus **Bus-** und **Sterntopologie**
- Über eine Sternstruktur sind die Stationen über ein HUB verbunden



Vermaschte Topologie

- **Dezentrales Netzwerk**
- Muss keinen verbindlichen Strukturen unterliegen
- Modell dient häufig als „**perfektes Netzwerk**“ indem jede Netzwerk-Station mit allen anderen Stationen verbunden ist
- Beim Ausfall einer Verbindung gibt es im zumeist **alternative Routen**
- Entspricht einem „kontrolliertem Chaos“ - das Internet stellt so ein gewolltes Netzwerk dar



+ Dezentrale Steuerung
+ Unendliche Netzausdehnung
+ Hohe Ausfallsicherheit

- Aufwendige Administration
- Teuere & hochwertige Vernetzung

Netzwerkgeräte - Netzwerkkarte

- Alternativ Netzwerkadapter oder **NIC** → **N**etwork **I**nterface **C**ard
- Ermöglicht den Zugriff auf ein Netzwerk
- Arbeitet auf dem **OSI Layer 1**
- Jede Netzwerkkarte hat eine **Hardwareadresse** die **weltweit eindeutig** ist (bzw. sein sollte)
 - → Beispielsweise: A4-23-7B-47-C1-66
- Diese Adresse wird als **MAC-Adresse** bezeichnet → Media Access Control
- Über die MAC lässt sich ein Client eindeutig identifizieren
- LEDs auf dem NIC zeigen den aktuellen Status (verbunden?) und die Datenübertragung



Netzwerkgeräte - Repeater

- **Kopplungselement**
- Frischt Datensignale auf = Verstärker
- Arbeitet auf **OSI Layer 1**
- Erhöht dadurch die Übertragungsstrecke
- Zumeist 2 Ports: Eingang und Ausgang
- **Arbeitet transparent**
 - Ist für andere Geräte nicht erken



Netzwerkgeräte - Hub

- Kopplungselement
- **Verbindet mehrere Stationen**
- Arbeitet auf **OSI Layer 1**
- Nimmt Datenpakete entgegen und **sendet diese an ALLE Ports**
- Somit „broadcasten“ Hubs
- Besitzt sog. „Uplink-Port“ um weitere Hubs zu verbinden



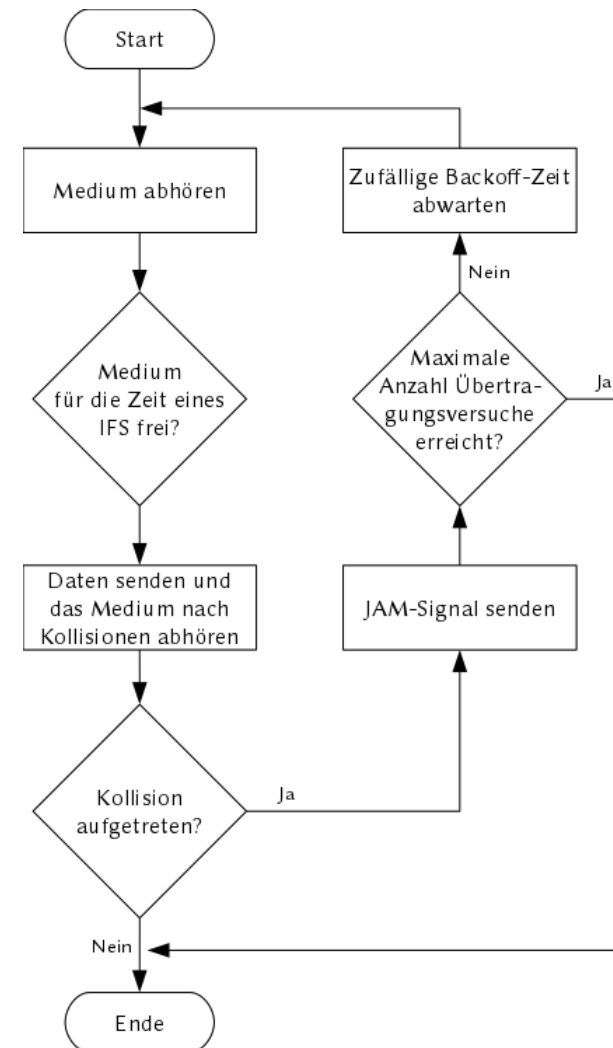
Netzwerkgeräte - Switch

- Kopplungselement
- Verbindet Stationen in einem Netzwerk
- Arbeitet auf **OSI Layer 2** – Data Layer
- Ähnlich einem HUB aber...
 - → Verbindungen werden direkt zwischen den Stationen geschaltet
 - → Sofern die zugehörigen Ports bekannt sind
 - → Verwendung von **MAC-Table** zur Zuordnung der Stationen
- Switch kann mit SPAN-Port zur Datenanalyse konfiguriert werden
 - → Switched Port Analyser (Mirror Port)



CSMA / CD

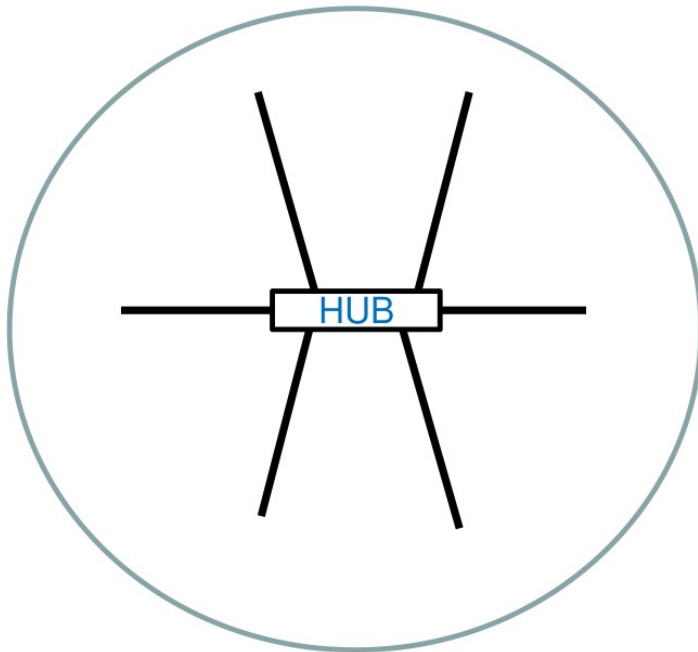
- **Carrier Sense Multiple Access Collision Detection**
- **Zugriffsverfahren**
- **Carrier Sense:**
 - Träger Zustandserkennung
- **Multiple Access:**
 - Mehrfachzugriff
- **Collision Detection:**
 - Kollisionserkennung
- **IFS = Interframe Spacing**
 - → zeitlichen Abstand zwischen gesendeten Paketen



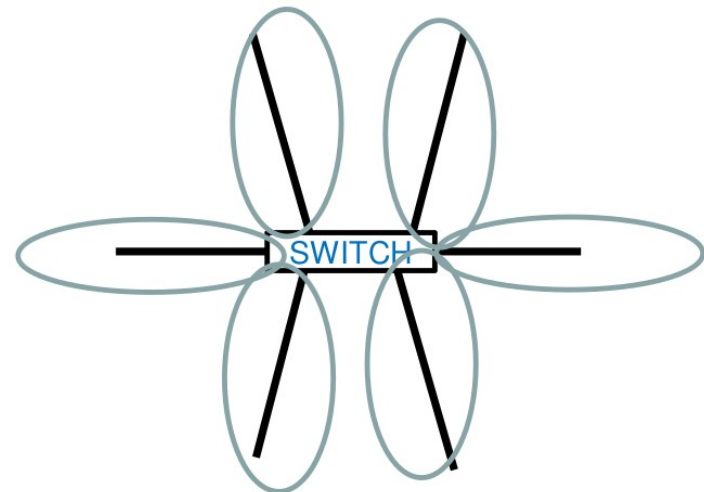
Kollisionsdomainen

Paketkollisionen entstehen durch Multiple Access (CSMA/CD)

Kollisionsdomäne



Kollisionsdomänen

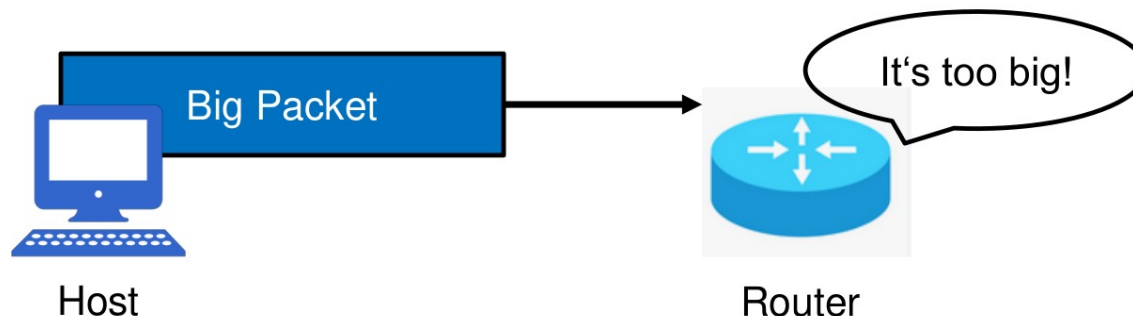


Router

- Verbinden Netze untereinander
- Arbeitet auf **OSI-Layer 3**
- Häufig an den Außengrenzen von Netzwerken
- Aufgaben
 - **Ermittlung** der verfügbaren **Routen**
 - **Auswahl** der geeignetsten **Route**
 - **Herstellen** der **Verbindung**
 - Anpassung der Datenpakete an Übertragungstechnik

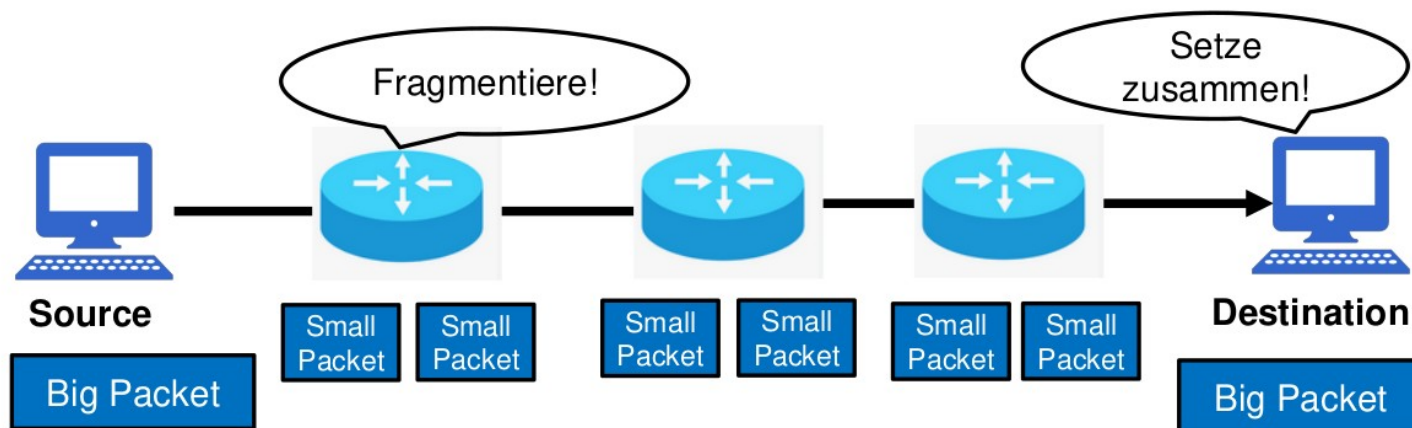
MTU Size

- MTU beschreibt die **maximale Paketgröße** eines Protokolls der **Vermittlungsschicht**
 - Fragmentierung (Aufteilung) eines Datenpaketes in mehrere physische Blöcke
 - findet statt, wenn die **Maximum Transmission Unit (MTU)** überschritten ist
 - Kann unterschiedliche Werte für unterschiedliche Protokolle annehmen



MTU Size / Fragmentierung

- Unterschiedliche Netzwerke haben **unterschiedliche MTUs**
 - Ethernet 1500 Bytes
 - Wifi 2300 Bytes
- Große MTUs sollten aus Effizienzgründen bevorzugt werden
- Schwierig da nicht jeder Router die gleichen MTU-Größen verwendet
- Zusammensetzung findet am Endclient statt (=Entlastung der Router)



MTU Size / Fragmentierung

- Router teilt Pakete die zu groß sind
 - Typischerweise in die größtmöglichen Paketteile
 - **Kopiert** den **IP Header** des Originals und **stattet damit die Paketteile aus**
 - Setzt ein **Offset** im IP-Header um die Fragmentposition anzugeben
 - Setzt das **MF-Flag** (**M**ore **F**ragments) im IP-Header bei jedem Fragment (außer das Letzte)
- Empfänger baut die Fragmente wieder zusammen
 - Identifiziert über das Identification Field welche Fragmente wie zusammengehören
 - Über das **MF-Flag 0** erkennt der Empfänger das letzte Fragment