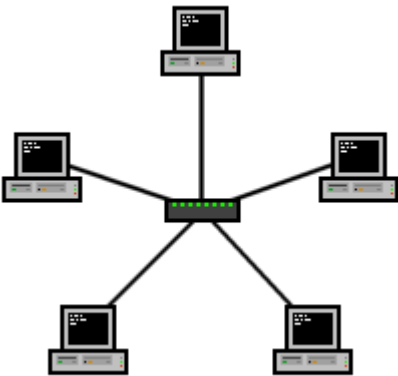


# Rechnernetze

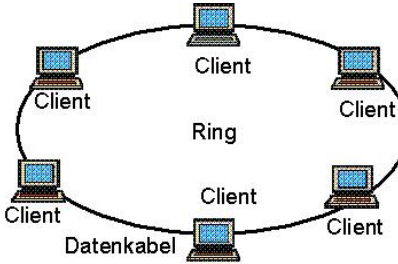
## Strukturen, Kommunikation und Protokolle


### Topologien

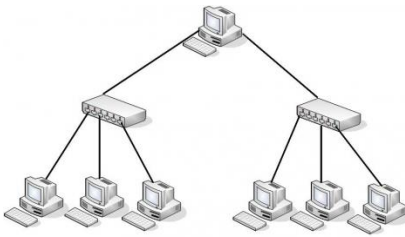
Unter einer Netzwerktopologie versteht man die Struktur, in der einzelne Netzwerkelemente miteinander verbunden sind. Diese ist insofern bedeutend, da von ihr u.a. die Leistungskapazität sowie die Stabilität des Netzwerks abhängig ist. Deshalb sollen im Folgenden die wichtigsten Topologien inklusive ihrer Vor- und Nachteile tabellarisch erfasst werden.

<u>Netzwerktopologie</u>	<u>Vorteile</u>	<u>Nachteile</u>
<b>Stern</b>	+	-
	Ausfall eines Endgeräts → keine Auswirkung auf Netzwerk	Ausfall des Verteilers – Netzverkehr unmöglich
	hohe Übertragungsraten, wenn Netzwerkknoten ein Switch	niedrige Übertragungsrate, wenn viele Hosts mit Hub verbunden
	leicht erweiterbar	hoher Verkabelungsaufwand
	leicht verständlich	
	leichte Fehlersuche	
	Multicast- / Broadcast- anwendungen gut möglich	
	kein Routing benötigt	

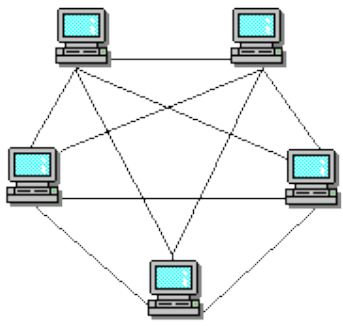
Bei kleineren Netzwerken oder auch Telefonanlagen ist die Sterntopologie Standard!

<b>Ring</b>	+	-
	keine Paketkollisionen – Vorgänger und Nachfolger sind definiert	hoher Zeitbedarf für Datenaustausch, speziell bei vielen angeschlossenen Komponenten
	alle Stationen arbeiten als Verstärker	teure Komponenten
	alle Rechner haben gleiche Zugriffsmöglichkeiten	
	garantierte Übertragungs- Bandbreite	relativ hohe maximale direkte Entfernung
	leicht erweiterbar	hoher Verkabelungsaufwand
	reguläre Topologie – jeder Client hat genau zwei Nachbarn – leicht programmierbar	Datenübertragungen können leicht abgehört werden
		nur ein Alternativweg bei Störung

Bus	+	-
	Ausfall eines Geräts hat keine Auswirkungen auf Netzwerk	Datenübertragungen sind unsicher
	geringe Kosten, da nur geringe Kabelmengen	Störung des Übertragungsmediums führt zum Ausfall
	einfache Verkabelung und Netzerweiterung	nur eine Station sendet, Rest blockiert
	keine aktiven Netzwerkkomponenten benötigt	bei Bussen mit Kollisionszulassung nur teils ausgelastet

Baum	+	-
	Ausfall eines Endgeräts → kein Netzwerkausfall	Ausfall eines Verteilers – die ausgehenden Geräte nicht mehr erreichbar
	strukturelle Erweiterbarkeit	ggf. Engpässe bei Kommunikation über Wurzel
	große Entfernungen realisierbar	große Bäume = schlechte Latenzen
	gute Eignung für Such- und Sortialgorithmen	

Die Baumtopologie ist eine erweiterte Sterntopologie! In den meisten Schulen wird diese Topologie verwendet. Der zentrale Server ist dabei direkt an der Wurzel des Baums angebunden. Von dort gehen Leitungen in die einzelnen Gebäude(teile). Innerhalb dieser Gebäude(teile) werden die einzelnen Netzwerkanschlüsse über weitere Switches angeschlossen.

Vermaschtes Netz	+	-
	sicherste Variante eines Rechnernetzes	viel Kabel ist notwendig
	Datenkommunikation auch bei Ausfall eines Endgerätes durch Umleitung möglich	ineffizient (Energie)
	leistungsfähig	komplexes Routing nötig für nicht vollvermaschte Netze
	vollvermaschte Netze benötigen kein Routing	

Das **Internet** ist in weiten Teilen ein **vermaschtes Netz**. Trotzdem gibt es "Hauptverkehrsadern" (die Backbone-Leitungen), die einem Bus ähneln. Die einzelnen Knoten stellen dabei die Schnittstellen (Router) zu den lokalen Netzwerken dar, die in einer eigenen Topologie realisiert sind.

## Das Client-Server-Prinzip

Die meisten Anwendungen im Internet basieren auf dem Client-Server-Prinzip.

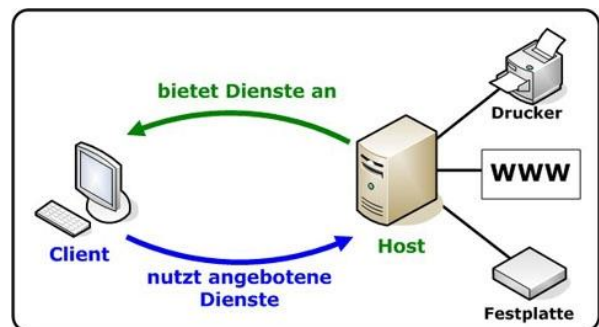
Ein **Server** (engl. „Diener“) ist ein Dienstleister, der in einem Computersystem Daten oder Ressourcen zur Verfügung stellt. Das Computersystem kann dabei aus einem einzelnen Computer oder einem Netzwerk mehrerer Computer bestehen. Zwei Bedeutungen werden unterschieden:

1. Server-Programm: Ein Computerprogramm, das einen Dienst (z. B. Fileserver: zentrale Speicherung von Dateien) bereitstellt.
2. Server-Computer: Der Computer, auf dem ein oder mehrere Server-Programme laufen. Die ursprüngliche Bezeichnung für diesen physischen Rechner ist **Host**.

Genauso gibt es für das Gegenstück, den **Client** (engl. „Kunde“), zwei Bedeutungen:

1. Ein Client ist eine Anwendung, die in einem Netzwerk den Dienst eines Servers in Anspruch nimmt.
2. Der Begriff Client wird aber auch oft verwendet, um einen Computer in einem Netzwerk zu bezeichnen.

Der Client (Rechner und Programm) ist bei einer Datenübertragung für die Kontaktaufnahme verantwortlich und bestimmt deren Zeitpunkt. Das hat für den Client-Rechner den Vorteil, dass er erst zum Zeitpunkt der Kontaktaufnahme eine Netzverbindung benötigt. Dies wird als **Client-Server-Prinzip** bezeichnet: Der "Kunde" (Client) sagt, was er will, der "Dienstleister" (Server) erbringt (daraufhin) die gewünschte Leistung.



Der **Webbrowser** ist ein Beispiel für einen Client, denn er sendet bei jedem Aufruf einer Webseite eine Anfrage an einen **Webserver** und erhält dann von diesem eine Antwort. Der Webserver macht die meiste Zeit nichts Anderes als warten. Er wird erst aktiv, wenn vom Client eine Anfrage eingeht.

Weitere interessante und viel genutzte Dienste werden z.B. von FTP-Servern, Print-Servern, Mail-Servern oder Proxy-Servern bereitgestellt.

Client-Server-Anwendungen sind heute Standard. Meist werden viele Server-Anwendungen auf einem (oder einigen wenigen) Rechner konzentriert (in den meisten Schulnetzen gibt es z.B. einen Server, der DHCP-Server, Fileserver, DNS-Server, Mailserver, Windows-Updateserver und vieles mehr gleichzeitig ist). Die Konzentration der Funktionalität auf den Server verringert den Administrationsaufwand und macht den einzelnen Nutzer unabhängig von einer bestimmten Hardware. So kann der Nutzer z.B. den Arbeitsort bzw. den Computer wechseln, ohne auf „seine“ Dateien etc. verzichten zu müssen.

Im Kontrast zum Client-Server-Modell steht das so genannte **Peer-to-Peer-Modell** (engl. peer: Gleichberechtigter). In Peer-to-Peer-Netzen kommunizieren die Rechner gleichwertig miteinander, d.h. jeder Rechner kann Dienste bereitstellen sowie nutzen und hat damit sowohl Server- als auch Client-Funktionen inne. Napster und Gnutella stellen recht bekannte Beispiele von P2P-Software dar.

Unter einem **Broadcast** (engl.; „Rundruf“) versteht man in einem lokalen Computernetzwerk eine Nachricht, bei der Datenpakete von einem Punkt aus an alle Teilnehmer übertragen werden, die dann selbst entscheiden, ob sie die Nachricht verarbeiten oder verwerfen möchten. Dies macht unter anderem dann Sinn, wenn die IP-Adresse des Empfängers einer Nachricht noch unbekannt ist. Konkrete Beispiele hierfür sind ARP (Zuordnung IP- zu MAC-Adressen), DHCP (zentrale Vergabe von IP-Adressen) und Wake On LAN.

## Das TCP/IP-Protokoll

Die oben beschriebene Kommunikation zwischen Client und Server kann natürlich nur dann funktionieren, wenn beide Parteien sich auf festgeschriebene, eindeutige Regeln (= Protokoll) einigen, welche das Format, den Inhalt, die Bedeutung und die Reihenfolge gesendeter Nachrichten festlegen.

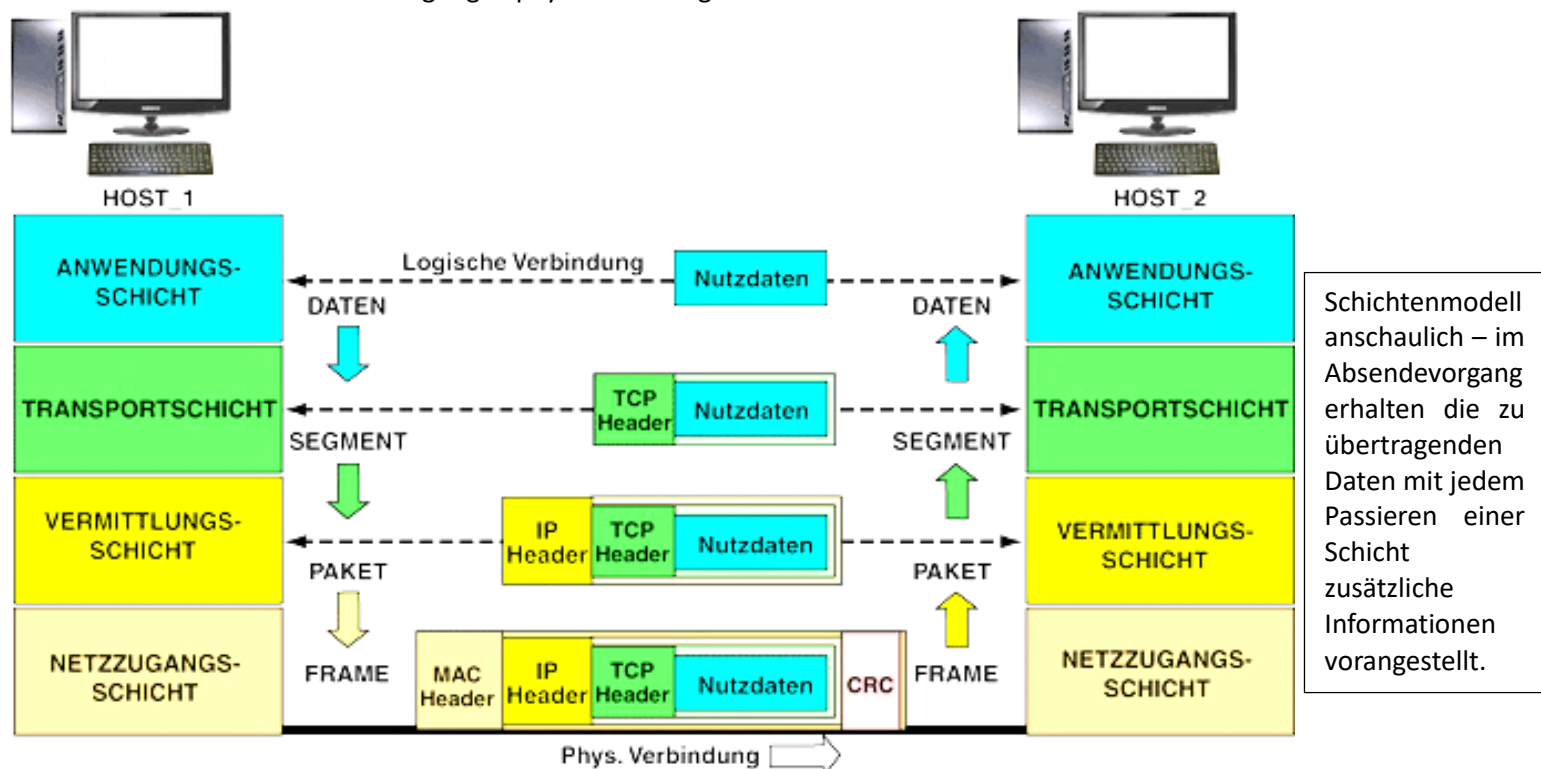
Im Folgenden werden wir nun das im Internet wichtigste Protokoll – das TCP/ IP-Protokoll – genauer unter die Lupe nehmen.

Die Tabelle rechts zeigt die vier dabei zugrundeliegenden Schichten sowie zugehörige Protokoll-Beispiele.

Die Aufgaben und Funktionen von TCP/ IP werden nun den jeweiligen Schichten zugeordnet:

TCP/IP-Schicht	Protokolle (Auswahl)
Anwendung	<a href="#">HTTP</a> <a href="#">FTP</a> <a href="#">SMTP</a> <a href="#">POP3</a> <a href="#">Telnet</a> <a href="#">DNS</a> <a href="#">SSH</a>
Transport	<a href="#">TCP</a> <a href="#">UDP</a>
Internet	<a href="#">IP (IPv4,IPv6)</a>
Netzzugang	<a href="#">Ethernet</a> , <a href="#">WLAN</a> , <a href="#">Token Ring</a>

- Anwendungsschicht:
  - Bereitstellung verschiedener Funktionalitäten wie Dateiübertragung (ftp), Übertragung von WWW-Seiten (http), E-Mail (smtp, pop3), Arbeit auf einem entfernten Rechner (telnet), usw.
  - Verbinden der zwei beteiligten Anwendungen, Festlegen von Checkpoints
  - Übersetzung der Daten in systemunabhängige Darstellung
- Transportschicht (TCP):
  - Herstellen einer zuverlässigen Verbindung zwischen den Computern
  - Aufteilen (Segmentierung) der zu übertragenden Daten in einzelne Pakete (um eine Überlastung zu verhindern) bzw. Zusammensetzen der Pakete
  - unabhängige Übertragung der einzelnen Pakete
  - Fehlerbehandlung (inkl. Zeitüberwachung)
  - Flusssteuerung (dynamische Auslastung der Übertragungsstrecke)
- Internetschicht (IP):
  - Logische Adressierung
  - Vermittlung der Pakete durch das Netzwerk (Routing)
- Netzzugangsschicht:
  - Herstellen zuverlässiger physikalischer Verbindung (Fehleranalyse durch Prüfsummen)
  - Adressumsetzung zwischen IP- und MAC-Adressen
  - Übertragung in physikalische Signale

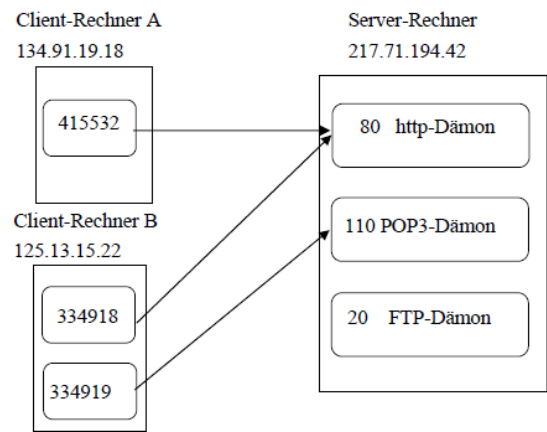


## TCP

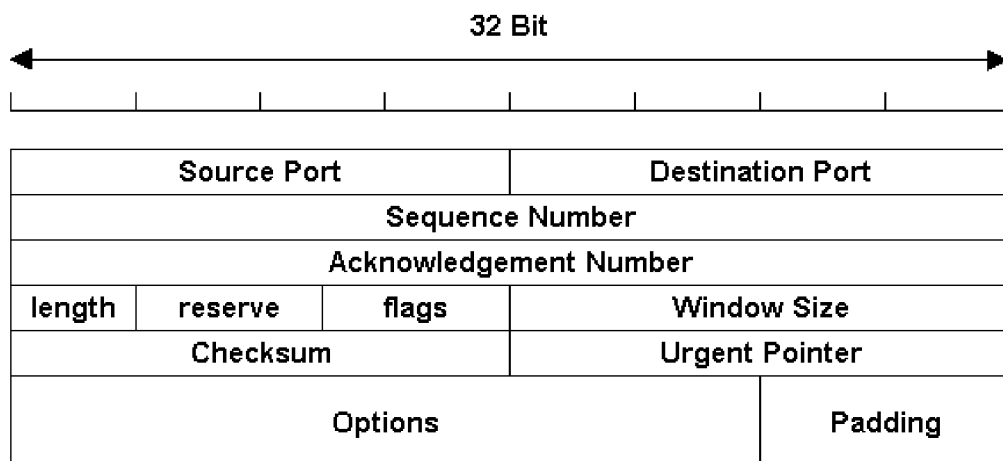
Die Aufgaben des TCP-Protokolls wurden ja bereits im vorangegangenen Textabschnitt erläutert. Der Ablauf der Übertragung soll allerdings nun ein wenig feiner betrachtet werden.

Zunächst einmal sei festgehalten, dass jede TCP-Verbindung eindeutig durch zwei Endpunkte identifiziert wird (den des Senders und den des Empfängers), wobei jeder Endpunkt ein geordnetes Paar, bestehend aus IP-Adresse und Port, darstellt. Ein solches Paar bildet eine bidirektionale Software-Schnittstelle und wird auch als Socket bezeichnet. Die Ports („=Eingänge“) sind dabei in jedem Rechner in der Regel für diverse Dienste vom Betriebssystem vordefiniert – Beispiele sind in der Tabelle rechts erkennbar.

Darüber hinaus sieht ein allgemeiner TCP-Header wie folgt aus:



Port	Protokolle, Dienste, Anwendungen,
20, 21	FTP: FTP Programme
22	SSH: sichere (Bank-) Verbindungen
23	Telnet: kleines Netzwerkprogramm
25	SMTP: Email Programme
53	DNS: Namensauflösung
80	HTTP: Internet Browser



### Erläuterungen:

- Sequence Number:** Bei jeder TCP-Verbindung werden Nummern zwischen den Kommunikationspartnern ausgehandelt. Während der Verbindung werden diese Nummern verwendet, um die TCP-Pakete eindeutig zu identifizieren.
- Ack. Number:** Alle Datenpakete werden bestätigt. Dazu dient das ACK-Flag und die „Acknowledgement“-Number, die sich aus der Sequence Number und der Anzahl der empfangenen Bytes errechnet. Damit kann der Sender feststellen, ob die Daten beim Empfänger vollständig angekommen sind.
- flags:** Kennzeichnung bestimmter für die Kommunikation und Weiterverarbeitung der Daten wichtiger Zustände (URG, ACK, PSH, RST, SYN, FIN).
- Window Size:** Der Empfänger sendet dem Sender in diesem Feld die Anzahl an Daten, die der Sender senden darf. Dadurch wird das Überlaufen des Empfangspuffers beim Empfänger verhindert. Den Vorgang nennt man Windowing und dient der Datenflusssteuerung.
- Urgent Pointen:** Zusammen mit der Sequence Number gibt dieser Wert die genaue Position der Daten im Datenstrom an. Der Wert ist nur gültig, wenn das URG-Flag gesetzt ist.

## IP

Aufgabe von IP ist die Zustellung der Datagramme im Netz – basierend auf den IP-Adressen von Sender bzw. Empfänger. IP kann mit dem Versenden eines Briefes per Post verglichen werden. Wenn der Brief aufgegeben wurde, hat der Sender keine Kontrolle mehr, ob dieser Brief wirklich ankommt. IP erhält die Segmente von TCP und ergänzt ebenfalls einen Nachrichtenkopf.

Ein allgemeiner IPv4-Header sieht dabei wie folgt aus:

4-bit	8-bit	16-bit	32-bit	
Ver.	Header Length	Type of Service	Total Length	
Identification			Flags	Offset
Time To Live	Protocol		Checksum	
Source Address				
Destination Address				
Options and Padding				

### Erläuterungen:

Version:	verwendete IP-Version-Nummer
Type of Service:	Der Type of Service (TOS) klassifiziert das Paket nach der Priorität der Zustellung.
Ident., Flags, Offset:	Diese Informationen steuern das Zusammensetzen der einzelnen übertragenen Pakete zu einer Einheit.
Time To Live:	Im Feld Time To Live (TTL) wird angegeben, nach wie vielen Routerdurchläufen das Paket verworfen wird.
Protocol:	Gibt das Protokoll an, das in der vorherigen Schicht verwendet wurde.

IP leitet die Daten aufgrund von Routing-Entscheidungen weiter. Zuerst wird mit Hilfe der Subnetzmaske entschieden, ob die IP-Adresse des Empfängers zum eigenen Netz gehört. Ist dies der Fall, werden die Daten unmittelbar an den Empfänger gesendet, andernfalls werden sie an den nächsten Router gesendet, der Verbindung zu diesem Zielnetz hat. Der Router, der das Datagramm erhält, prüft ebenfalls, ob er mit dem Zielnetz direkt verbunden ist und das Datenpaket direkt zustellen kann. Wenn nicht, leitet er es, entsprechend seiner Routing-Tabelle an einen weiteren Router weiter. So können Datagramme über viele Router geleitet werden, bis sie schließlich den Zielrechner erreichen. Dabei kann es durchaus passieren, dass Datenpakete fehlgeleitet werden und im Netz umherirren. Damit diese das Netz jedoch nicht dauerhaft belasten, gibt es das so genannte TTL-Attribut im IP-Header (siehe oben). Die IP-Schicht des Empfängers prüft schließlich die Datagramme auf Korrektheit, entfernt den Nachrichtenkopf und gibt die Segmente an die TCP-Schicht weiter.

### Subnetzmaske 255.255.255.0

IP-Adresse 192.168.178.125



**Subnetzmaske:** Die ersten 3 Zahlenblöcke

**Netzwerkennung:** 192.168.178.0

**Netzwerk-ID:** 192.168.178.

**Host-ID:** 125

**Zum Netzwerk gehören:** 192.168.178.1 – 192.168.178.254