FI

Teil 3

Netzwerktechnik

klin, 01.03.2022

Inhalt

1 Netzwerktechnik

1.1 Allgemeines

1.1.1 Definition

Ein Netzwerk ist eine Infrastruktur, die Datenendgeräten die Kommunikation, den Datenaustausch und die Nutzung gemeinsamer Ressourcen **transparent** ermöglicht. Transparent bedeutet, dass sich der Endbenutzer muss sich nicht darum kümmern, mithilfe welcher Verfahren, Geräte und Medien die Informationen transportiert werden.

1.1.2 Geschichte

Die Geschichte der elektronischen Kommunikationstechnik beginnt im 20. Jahrhundert mit der Telegrafie und dem Rundfunk. **Nach dem Entstehen der ersten Computer** Mitte des 20. Jahrhunderts tauchten bald die ersten Ansätze der digitalen Datenkommunikation auf.

Ein wesentlicher Schritt war die Einführung der **paketorientierten Kommunikation** im Unterschied zu der bis dato gängigen **leitungsorientierten Kommunikation**. Weitere Meilensteine waren die Entwicklung des **TCP/IP Protokolls**, das Aufkommen der **Personal Computer** und die rasante Entwicklung des **Internets**.

Vorteile der **paketorientierten Datenübertragung**:

- Hohe Übertragungsraten durch effiziente Leitungsnutzung (Zeitmultiplex)
- Hohe Zuverlässigkeit durch Wiederholung oder Korrektur fehlerhaft übertragener Pakete

1.1.3 Nutzen von Netzwerken

Im Bereich der mittleren und großen Rechenanlagen war Vernetzung relativ früh schon ein Thema. Der PC wurde zunächst als typisch nicht netzwerkfähiges Gerät konzipiert und entwickelt. Doch auch hier zeigten sich bald die Notwendigkeit und der Nutzen der Vernetzung.

- Datenaustausch zwischen den Teilnehmern
- Zugriff auf gemeinsam genutzte Daten (Datenspeicherung, Versionsverwaltung)
- Gemeinsame Nutzung teurer Ressourcen (Laserdrucker)
- Nutzung von Internet- und Intranet-Diensten

1.1.4 Hard- und Softwarekomponenten von Netzwerken

- kommunikationsfähige Endgeräte
- Übertragungsmedien (Kabel, Funk,..)
- Übertragungseinrichtungen (Netzwerkskarten, Switch, Router, Gateways)
- Protokolle zum Datentransport
- Dienste zur Nutzung in Anwendungen

1.1.5 Netzwerkstrukturen

Man unterscheidet zwei Arten, Netzwerke zu strukturieren:

- **Peer-to-Peer Struktur**:
  P2P-Netzwerke, sind Netzwerke, bei denen die Teilnehmer direkt miteinander verbunden sind und die gleichen Rechte besitzen. Das bedeutet, dass die Kommunikation nicht über einen Server erfolgt, sondern direkt von einem Computer zu einem anderen Computer.
  Filesharing-Dienste wie BitTorrent nutzen Peer-to-Peer**.** Dateien werden dabei zwischen den Rechnern einzelner Nutzer gesucht und direkt geteilt.
- **Client-Server Struktur**:
  Das Client-Server-Modell ist ein Konzept zur Verteilung von Diensten und Aufgaben in einem Netzwerk. Dienste werden von Servern bereitgestellt und können von Clients genutzt werden. Eine typsche Anwendungen des Client-Server-Modells ist der **Zugriff auf Webseiten** über das HTTP-Protokoll.

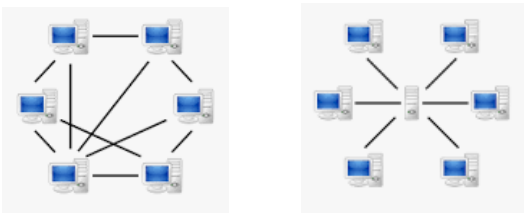


Abb. 1 P2P- und Client-Server Struktur

1.1.6 Netzkategorien

Rechnernetze werden unterschieden in Lokalen Netzen und Weitverkehrsnetzen.

- **Local Area Network (LAN):**
  Rechner einer Abteilung oder einer Firma; lokal organisiert; höchste Übertragungsraten; Ausdehnung einige 100 Meter; Topologie Bus, Stern, Ring; Ethernet Technologie
- **Wide Area Networks (WAN):**
  Weltweite Verbindungen; dezentral organisiert; Datenraten meist kleiner als bei LAN; Standleitungen, Satellitenkanäle, Richtfunk, Tiefseekabel; vermaschte redundante Netzwerke

1.2 Internet

Das Internet ist das **weltweit größte Netz** aus miteinander verbundenen Computernetzen, die das **TCP/IP-Protokoll** benutzen. Durch die Verwendung des TCP/IP-Protokolls können Teilnehmer plattformunabhängige Dienste wie E-Mail, FTP, HTTP, usw. in Anspruch nehmen.

Kennzeichen der Internet-Technologie:

- globales Adressierungsschema (IP-Adresse)
- TCP/IP-Protokolle + Routing-Verfahren + Hilfsdienste
- Anwendungsprotokolle: email, ftp, telnet, http, ...

Entwicklungsgeschichte:

- 1970er Jahre: Pionierzeit; 1969 ersten Netzwerk aus 4 Universitäten, 1974 TCP/IP
- 1980er Jahre: Wissenschaftsnetz; erste europäische Universitäten schließen sich an
- 1990er Jahre: Kommerzialisierung

1.2.1 Standardisierung im Internet

Die **Internet SOCiety (ISOC)** wurde 1992 gegründet und ist als Nichtregierungsorganisation für die Pflege und Weiterentwicklung der Internetinfrastruktur zuständig. Die ISOC beherbergt die für die Internetstandards zuständigen Gremien. Durch regelmäßige Veröffentlichungen werden aktuelle Entwicklungen vorgestellt.

Homepage: https://www.internetsociety.org/

Die **Internet Assigned Numbers Authority (IANA)** ist ein Gremium der ISOC und für die Zuordnung von Nummern und Namen im Internet, insbesondere von IP-Adressen, zuständig. Sie ist eine der ältesten Institutionen im Internet.

Homepage: https://www.iana.org

1.2.2 RFCs - Request for Comments

Eine wichtige Rolle bei der Entstehung und Entwicklung des Internet spielen die sogenannten **Request for Comments (RFCs)**. RFCs sind Dokumente, in denen die **Standards für das Internet** veröffentlicht werden. Einige **RFCs beschreiben Dienste und Protokolle sowie deren Realisierung**, andere fassen Regeln und Grundsätze (policies) zusammen.

Ist ein Dokument veröffentlicht, wird ihm eine **RFC-Nummer** zugewiesen. Die Dokumente werden von einer Arbeitsgruppe und/oder dem **RFC-Editor** geprüft. Dabei durchläuft das Dokument verschiedene Stufen, die Stufen der Entwicklung, Testung und Akzeptanz.

Protokoll Register: https://www.iana.org/protocols

1.3 Intranet

Ein Intranet (lat. intra "innerhalb") ist ein Rechnernetz, das im Gegensatz zum Internet unabhängig vom öffentlichen Netz benutzt werden kann und **nicht öffentlich zugänglich** ist.

Im Unterschied zu den Netzkategorien **LAN** und **WAN** bezeichnet das Intranet nicht die räumliche Ausdehnung eines Rechnernetzes, sondern den **begrenzten Benutzerkreis**.

1.4 Ethernet

Ethernet ist die am weitesten verbreitete Technologie **für Local Area Networks**. Ethernet wurde von den Firmen **DEC, Intel** und **Xerox** (DIX-Standard) ab **1976** entwickelt. Der Ethernet-Standard umfasst die **Schichten 1** und **2** (Bitübertragungs- und Sicherungsschicht) des OSI-Referenzmodells.

Der Standard enthält Vorschriften über **Netzwerkarchitektur**, **Zugriffsverfahren**, **Übertragungs-verfahren** und **Hardware**. Es handelt sich dabei also nicht um ein vollständiges Netzwerkprotokoll, sondern ein Subnetz, in dem andere Protokolle arbeiten können, wie z.B. TCP/IP.

Die **Arbeitsgruppe 802** des *Institute for Electrical and Electronic Engineers* (IEEE) hat für Ethernet in LANs einige Standards festgelegt. **Ethernet ist seit 1983** in der IEEE-Norm 802.3 **standardisiert**.

<table>
<tr><td rowspan="2">2</td><td rowspan="3">802.1
Internet-Working</td><td colspan="4">802.2
Logical Link Control (LLC)</td></tr>
<tr><td colspan="4">802.1
Media Access Control (MAC)</td></tr>
<tr><td>1</td><td>802.3
Ethernet</td><td>802.4
Token-Bus</td><td>802.5
Token-Ring</td><td>802.11
Wireless LAN</td></tr>
</table>

Abbildung 1 Standardisierung nach IEEE 802

1.4.1 Funktionen des Ethernet-Protokolls

- Bereitstellung der **Bitübertragungsschicht – Schicht 1**
  - Senden und empfangen serieller Bitströme über das Medium
  - Detektieren von Kollisionen (bei Half-Duplex Betrieb ohne Switch)
- Bereitstellung der **Sicherungsschicht** mit **pysikalischen Adressierung – Schicht 2**
  - Logical Link Control (LLC)
    - Bereitstellen von Datenkanälen für darüberliegende Anwendungen
  - Media Access Control (MAC)
    - Regelt den Zugriff auf die unterschiedlichen Übertragungsmedien

1.4.2 Aktuelle Ethernet-Technologien

- **Fast Ethernet 100Base-TX** (**Standard in der Feldbustechnik**, z.B. Profinet)
  100MBit/s, Kupfer Twisted Pair , 100m, CAT5, 125MBaud
  2 Adernpaare, jeweils eines zum Senden und Empfangen (Vollduplexbetrieb)
- Gigabit Ethernet **1000Base-T**
  1GBit/s, Kupfer **T**wisted Pair, 100m, CAT5, 125MBaud, PAM5-Kodierung
  4 Adernpaare zum simultanen Senden und Empfangen (Vollduplexbetrieb, **Echo-Cancellation**)
- Gigabit Ethernet **1000Base-SX** und **1000Base-LX**
  1GBit/s, Glasfaser (SX=Multimode / LX=Singlemode), bis 5000m mit Singlemode

1.4.3 Auto-Negotiation

Auto Negotiation (deutsch: Aushandlung) bezeichnet ein Verfahren, das es zwei miteinander verbundenen Ethernet-Netzwerk-Ports erlaubt, selbständig die Kommunikationsparameter wie maximal mögliche **Übertragungsgeschwindigkeit** und das **Duplex-Verfahren** (Half- oder Full-Duplex) miteinander auszuhandeln und zu konfigurieren.

1.5 TCP/IP Protokollfamilie

Transmission Control Protocol / Internet Protocol (TCP/IP) ist eine **Protokollfamilie**, die für die Kommunikation über große Netzwerke, die aus verschiedenen über Router verbundenen Netzwerksegmenten bestehen, entwickelt wurde.

Die Entwicklung von TCP/IP hatte ein Ziel vor Augen:
Das Protokoll sollte besonders **robust** und **fehlertolerant** sein. Die verbundenen Computer sollten auch noch kommunizieren können, wenn ein Rechner oder eine Verbindung ausfällt.

Diese Idee erkennt man an folgenden zwei **Eigenschaften des Internet:**

- Das Netz ist **nicht hierarchisch** aufgebaut. Es gibt keine oberste Instanz oder zentralen Knoten, wo alle Daten durch müssen.
- Die Daten werden im Internet **paketweise** transportiert. Eine lange Botschaft wird in viele kleine nummerierte Pakete verpackt und verschickt. Nicht alle Pakete müssen den gleichen Weg zum Empfänger nehmen. Jedes Paket sucht sich den Weg, der gerade am wenigsten Verkehr hat. Der Empfänger packt dann die Pakete in der richtigen Reihenfolge wieder aus.

Das hat verschiedene Vorteile:

- Die zu transportierenden Daten werden gleichmäßig auf die vorhandenen Leitungen verteilt
- Die Leitung wird nur während des Transports der Pakete belegt
- Im Fehlerfall muss nur ein kleiner Teil der Nachricht, nämlich das defekte Paket, erneut versandt werden.

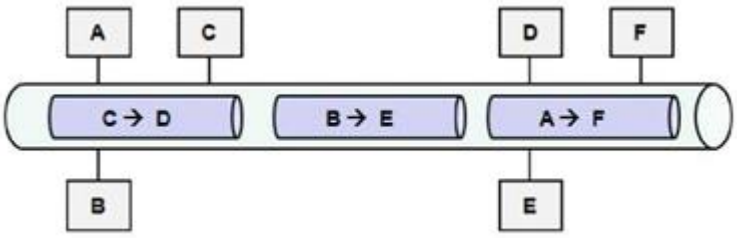


Abbildung 2 Paketvermittlung

Die TCP/IP-Familie kann perfekt im OSI-Modell abgebildet werden. Zur Vorstellung der TCP/IP-Familie wird ein **vereinfachtes, vierschichtiges TCP/IP-Modell** verwendet.

1.6 Schichtenmodell

Netzwerkprotokolle werden normalerweise in **Schichten** entwickelt, wobei jede Schicht für einen anderen Aspekt der Kommunikation zuständig ist.

Auf ihrem Weg von einer Anwendung zur anderen durchlaufen die Daten verschiedene Schichten. Jede dieser Schichten übernimmt dabei eine andere Funktion, auf die die nächsthöhere Schicht wiederum aufbaut.

TCP/IP ist die Kombination aus verschiedenen Protokollen in verschiedenen Schichten. TCP/IP wird in der Regel als 4-Schichtensystem betrachtet. Die Schichten 5 und 6 (Sitzung und Darstellung) entfallen.

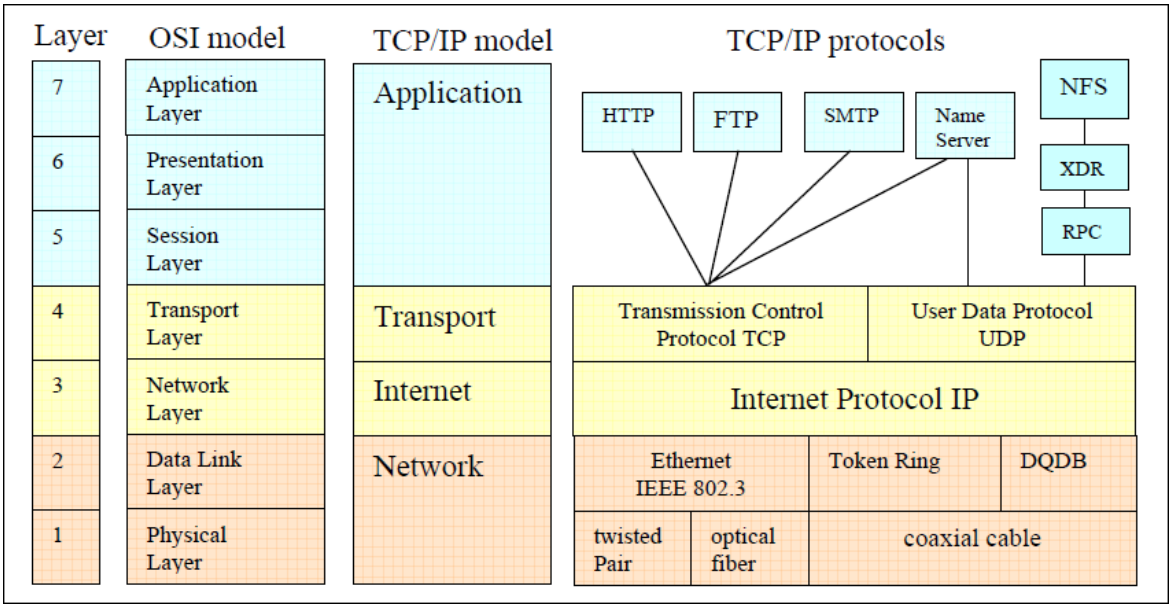


Abbildung 3 Die 4 Schichten der TCP/IP Protokollfamilie

TCP/IP liefert das Datenpaket schließlich nicht nur beim richtigen Empfänger, sondern auch bei der richtigen Applikation ab, nämlich einem weiteren übergeordneten Protokoll, welches mit einem Anwendungsprogramm zusammenarbeitet.

1.6.1 Adressierung im Internet

Im Internet existieren in den unterschiedlichen Ebenen drei **unabhängige Adressierungssarten**:

- **Physikalische Adressen**
  Ethernet- oder MAC-Adresse, z.B. 18:5E:0F:30:B0:56 – **in Schicht 1, 2**
- **Internet Adressen**
  IP-Adresse, z.B. 192.168.2.1 – **Schicht 3**
- **Domain Namen**
  z.B. www.litec.ac.at – **Schicht 7**

1.6.2 Verkapselung

Wenn eine Anwendung Daten mit Hilfe von TCP sendet, werden die Daten im **Protokoll-Stack** durch die Schichten nach unten gereicht, bis sie als Bitstrom über das Netzwerk übertragen werden können. Jede Schicht fügt den Daten weitere Informationen hinzu, indem sie den übergebenen **Daten Header** voranstellt (und manchmal Trailer hinten anhängt).

Die Dateneinheit, die TCP an IP sendet, wird **TCP-Segment** genannt. Die Dateneinheit, die IP an die Netzwerkschnittstelle sendet, heißt **IP-Datagramm**. Der Bitstrom, der über das Ethernet fließt, wird **Ethernet-Frame** genannt.

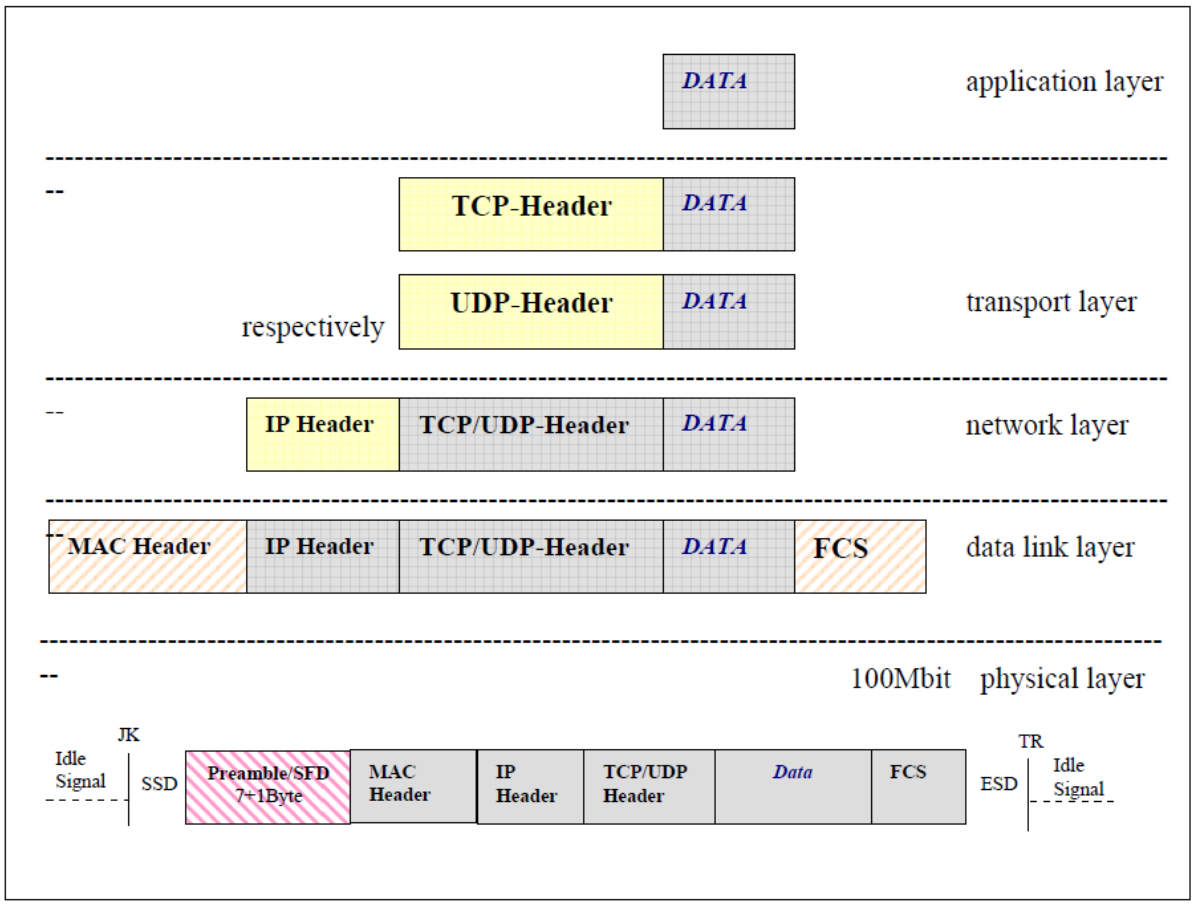


Abbildung 4 Verkapselung der Daten im TCP/IP Protokoll Stack

2 Sicherungsschicht, Layer 2

2.1 MAC-Adresse

Auf einem gemeinsamen Übertragungsmedium eines LAN muss jede Station über eine eindeutige Adresse verfügen. Jeder Teilnehmer hat eine Ethernet-Adresse, eine **physikalische Adresse**, die der Netzwerkkarte zugeordnet ist: die **MAC-Adresse** (**Medium Access Control Address**).
Jeder Hersteller von Netzwerkkarten gibt jeder Karte eine einzigartige Adressnummer, die im ROM der Karte gespeichert wird.

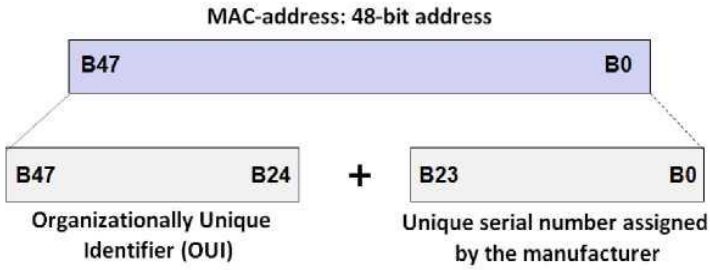


Abbildung 5 Die MAC-Adresse

Die MAC-Adresse setzt sich aus **48 Bits** (**6 Bytes**) und ist in **zwei Gruppen zu je drei Bytes** aufgeteilt. Die höherwertigen 24 Bits bilden eine **Herstellernummer**. Die niederwertigsten 24 Bits bilden eine **Seriennummer**. **Jede MAC-Adresse darf weltweit nur einmal vergeben werden.** Eine Broadcast Sendung in der Broadcast Domäne erfolgt mit der **Broadcast MAC-Adresse** FF:FF:FF:FF:FF:FF.

2.2 Ethernet-Frame

Das Ethernet-Frame muss zur zuverlässigen Kollisionserkennung bei max. Leitungslänge und gegebener Laufzeit eine **Mindestlänge von 64 Byte** haben. Ein Ethernet-Frame besteht aus mindestens 46 effektiven Datenbytes und einer konstanten Anzahl von 18 Protokollbytes (ohne Präambel). Diese Mindestanzahl an Datenbytes ist aufgrund der Definition der Slot-Zeit nötig.

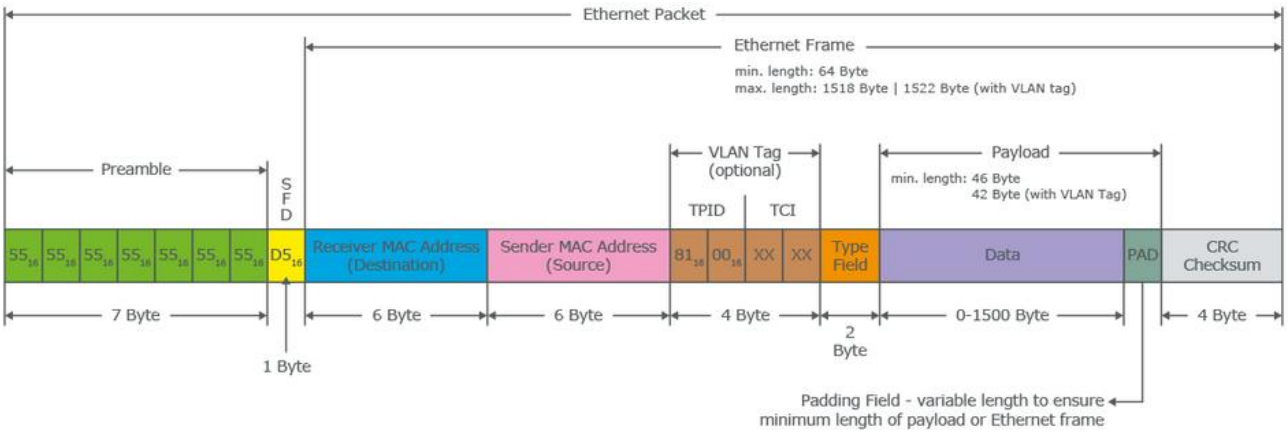


Abbildung 6 Ethernet Frameaufbau nach IEEE 802.3

Ethernet-Frame Aufbau:

- **Präambel** 56 Bits jeweils abwechselnd 1 und 0. Diese Bits werden für die Synchronisation verwendet und verschaffen jedem Teilnehmer die nötige Zeit, um die Aktivität auf dem Bus wahrzunehmen, bevor die Nutzdaten übertragen werden.
- **Start of Frame Delimiter** (SFD) ist das letzte Byte der Präambel und signalisiert dem Empfänger, dass als nächstes die Nutzdaten folgen.
- **Destination Adresse** (DA) identifiziert die Station/Stationen, für die die Nachricht bestimmt ist/sind. Die Zieladresse kann individuell, eine Multicast- oder eine Broadcast-Adresse sein.
- **Source Adresse** (SA) identifiziert die Station, von der die Nachricht stammt.
- **VLAN-Tag** (optional) dient zur Unterteilung eines einzelnen physischen Netzwerks in mehrere logische Netzwerke.
- **TYP-Feld** gibt Auskunft über das verwendete Protokoll der nächsthöheren Schicht innerhalb der Nutzdaten. z.B.:
    - 0800h IP (Internet Protocol)
    - 0806h ARP Protocol
    - 08892 Echtzeit Ethernet Profinet
    - 088A4 Echtzeit Ethernet EtherCat
- **DATA** Datenfeld mit frei wählbaren Nutzdaten. Die Länge muss mindestens 46 und darf höchstens 1500 Bytes betragen.
- **Padding-Bytes** (PAD) sind willkürliche Daten-Bits, um das Ethernet-Frame auf die Mindestlänge von 64 Byte zu vervollständigen.
- **CRC-Prüfsumme** die vom Sender errechnet und mitgesendet wird.

Die **Maximum Transmission Unit MTU** (maximale Übertragungseinheit) beschreibt die maximale Nutzdatenlänge eines Protokolls, welches ohne Fragmentierung von der Vermittlungsschicht (Schicht 3) übertragen werden kann. Bei Ethernet beträgt die MTU **1500 Byte**.

2.3 Address Resolution Protokoll (ARP)

Will ein Rechner A an einen Rechner B Daten versenden, muss er die **physikalische Adresse** des Empfängers **ermitteln**. Ohne diese kann er keine definitive Empfängeradresse angeben, und ohne diese weiß kein Rechner, dass die Daten für ihn bestimmt sind. Bekannt ist erst einmal nur die logische Adresse des Empfängers.

Der sendende Rechner sendet also als Erstes einen **ARP-Request** an alle Rechner. Er sendet seine logische und physikalische Adresse, die **logische Adresse des Empfängers** an die **Broadcast-Adresse FF:FF:FF:FF:FF:FF**. Alle Rechner des Netzes empfangen das Paket und sehen nach, ob ihre logische Adresse darin enthalten ist. Wenn nicht, wird das Paket verworfen. Wenn ja, merkt sich der Empfänger die Daten des Senders, trägt seine physikalische Adresse in das Paket ein und sendet es zurück. Sender und Empfänger kennen nun die zur logischen Adresse gehörige physikalische Adresse des Partners und können miteinander kommunizieren.

Auch für die Umkehrfunktion gibt es eine standardisierte Vorgehensweise, den **Reverse-ARP RARP**. Hier sendet die Station A unter Angabe ihrer physikalischen Adresse einen **RARP-Request**.

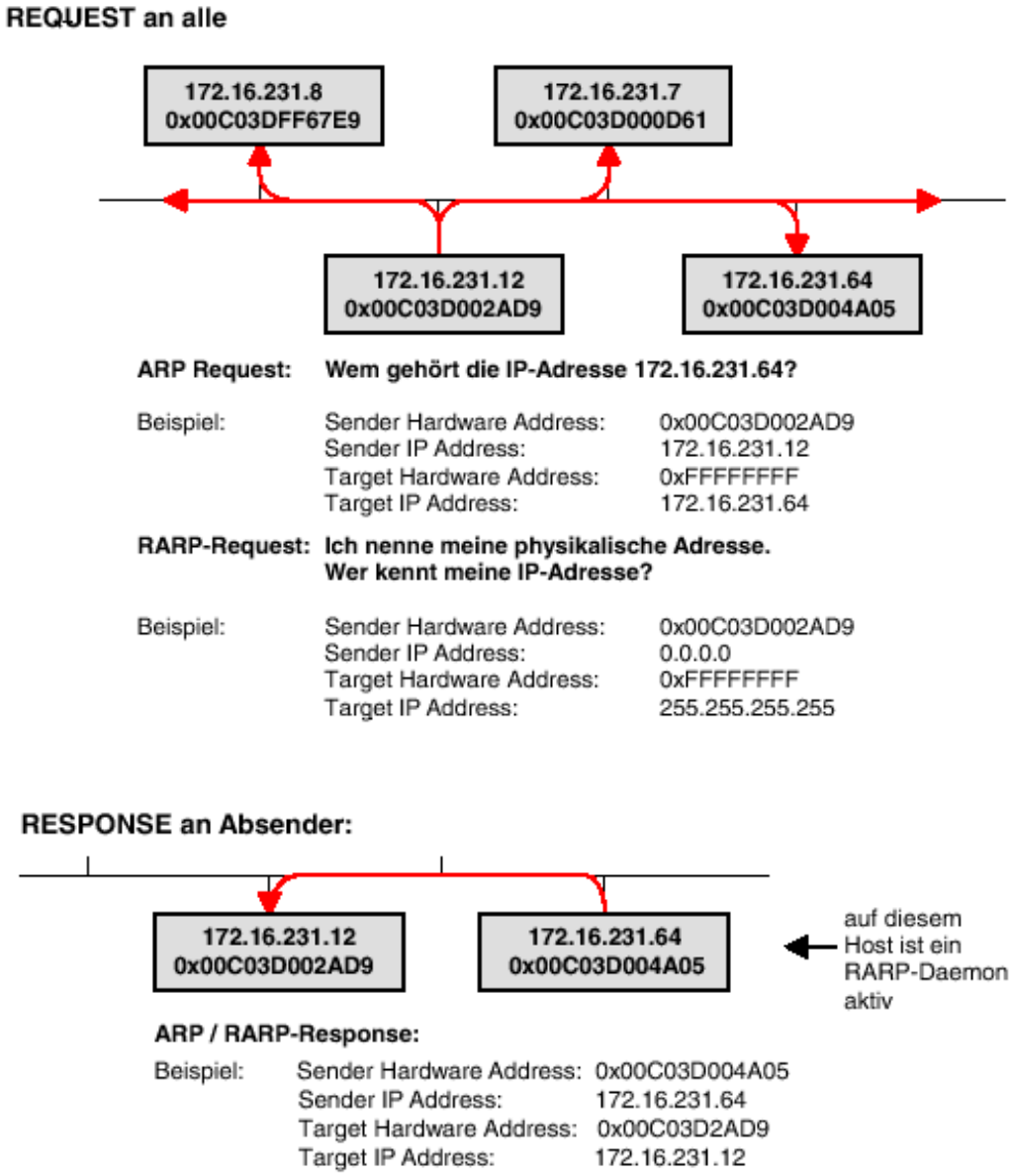


Abbildung 7 ARP-Request / Response

ARP-Cache:

300 Sekunden nach dem letzten Kontakt verfallen diese Informationen. Dies ist ein empfohlener Richtwert, hier ist alles nicht so ganz starr implementiert und stark vom Hersteller abhängig, gültig ist eine Zeit von 10 bis 10 Millionen Sekunden. Nach dieser Zeit müssen wieder ARP Anfragepakete ausgetauscht werden. Der Speicher, in dem die Daten zwischengespeichert werden, nennt sich ARP-Cache.

Durch dieses Verfahren ist sichergestellt, dass ein Netzwerkgerät nach Änderung seiner logischen Adresse ohne irgendwelche Konfigurationen erreichbar bleibt. Analog gilt dasselbe auch für eine Änderung der physikalischen Adresse, zum Beispiel nach Austausch einer defekten Netzwerkkarte.

Konsolenkommando *arp*:

Das Kommando ***arp -a [Inet-Adresse]*** schreibt den Inhalt des ARP-Cache in die Konsole. Falls Inet-Adresse angegeben wurde, werden die IP und zugehörige MAC-Adresse angezeigt. ***arp -d*** löscht alle Einträge im ARP-Cache. ***arp -s Inet-Adresse*** erzeugt einen statischen Eintrag und verbindet manuell eine IP-Adresse mit einer MAC-Adresse.

```
c:\Users>arp -a 192.168.2.1

Schnittstelle: 192.168.2.100 --- 0x13
  Internetadresse       Physische Adresse     Typ
  192.168.2.1           60-e3-27-be-17-f4     dynamisch
```

Konsolenkommando *getmac*:

getmac zeigt die MAC-Adressen aller Netzwerkadapter des Computers an.

```
c:\Users>getmac -v

Verbindungsname Netzwerkadapter Physisch. Adresse   Transportname
=============== =============== =================== ==================
WLAN            Intel(R) Dual B 18-5E-0F-30-B0-56   Nicht zutreffend
Ethernet        Intel(R) Ethern F8-CA-B8-2C-41-03   Medien ausgeworfen
Bluetooth-Netzw Bluetooth Devic 18-5E-0F-30-B0-5A   Medien ausgeworfen
VMware Network  VMware Virtual  00-50-56-C0-00-01   Nicht zutreffend
VMware Network  VMware Virtual  00-50-56-C0-00-08   Nicht zutreffend
```

3 Vermittlungsschicht, Layer 3

Diese Vermittlungsschicht ist für das Transportieren und Bereitstellen von Informationen über verschiedene Netzwerke verantwortlich.

3.1 Das IP-Protokoll

Aufgaben des IP-Protokolls:

- **Übertragung von Datenpaketen** der Transportschicht **zwischen zwei Netzwerkknoten**
- **Unterteilen von zu großen Datenpaketen** in übertragbare durchnummerierte Pakete (es erfolgt keine gesicherte Übertragung, der Verlust von Paketen ist möglich).
- **Transportieren der Datenpakete über Netzwerke unterschiedlicher Technologien,** wie z.B. über Ethernet- oder Serielle-Verbindungen (auch mit unterschiedlichen Paketgrößen).
- **Dynamisches Routing** mittels Routing-Tabellen (Pakete einer Datenübertragung können unterschiedliche Wege nehmen).

3.2 IP V4 Adressierung

Zum Transport ist eine **einheitliche Adressierung** nötig. Wenn verschiedene Netzwerke zu einem großen Ganzen verbunden werden, dann muss auch jedes Netzwerk an einer eindeutigen Adresse identifizierbar sein. Deshalb erhält jedes Netzwerk eine **eindeutige Netzwerkadresse**. Ausgehend von dieser Netzwerkadresse wird jedem Teilnehmer (Hosts und Router) des Netzwerks eine **eindeutige IP-Adresse** innerhalb dieses Netzwerkadressraums zugeteilt.

Die IP-Adresse ist bei **IP V4** eine 32-bit lange Zahl. Die 32 Bit werden in 4 Bytes zu jeweils 8 Bit zusammengefasst, denen dann die entsprechende Dezimalzahl zugeordnet wird. Diese **Oktetts** (Wertebereich 0 bis 255) werden durch Punkte voneinander getrennt.

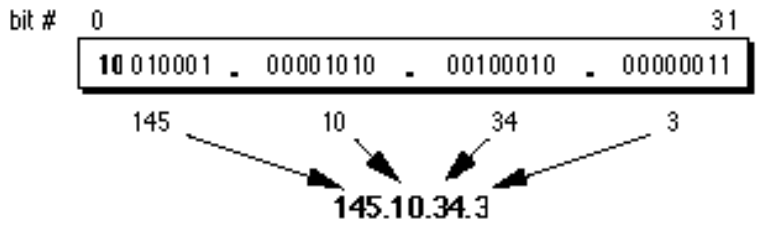


Abbildung 8 IP V4 Adresse

Jede IP-Adresse besteht aus zwei Teilen. Der Bereich für die Adressierung des Netzwerk liegt in den vorderen Oktetts (**Netzanteil** od. **Net-ID**) und der Bereich für den Host in den restlichen Oktetts (=**Hostanteil** od. **Host-ID**).

Die IP-Adressen werden weltweit koordiniert von der IANA (Internet Assigned Numbers Authority) vergeben. Da es verschieden große Netze gibt, wurden sogenannte Adressklassen geschaffen.

Es gibt **fünf Adressklassen**:

- Class-A Adressen: große Netze mit sehr vielen Hosts
- Class-B Adressen: für mittelgroße Netze
- Class-C Adressen: kleine lokale Netze mit wenigen Hosts
- Class-D Adressen: für Multicast Adressen
- Class-E Adressen: für zukünftige Anwendungen reserviert

Class	1.Oktett					2.Oktett	3.Oktett	4.Oktett
A	0	Net-ID				Host-ID		
B	1	0	Net-ID				Host-ID	
C	1	1	0	Net-ID				Host-ID
D	1	1	1	0	Multicast Adressen			
E	1	1	1	1	0	reserviert		

Abbildung 9 Die 5 Adressklassen

Class	Bereich	Netze	Hosts	Anteil
A	**1.0.0.0 - 127.**255.255.255	126 = 2^7- 2	16777214 = 2^{24}- 2	50,00%
B	**128.0.0.0 - 191.255.**255.255	16382 = 2^{14}- 2	65534 = 2^{16}- 2	25,00%
C	**192.0.0.0 - 223.255.255.**255	2097150 = 2^{21}- 2	254 = 2^8- 2	12,50%
D	**224.0.0.0 - 239.255.255.255**			6,25%
E	**240.0.0.0 - 255.255.255.255**			6,25%

Abbildung 10 IP Adressbereiche

Die Anzahl der Netze ist jeweils um 2 reduziert, weil eine Netzwerkadresse bei der alle Bits auf 1 oder 0 gesetzt sind ungültig ist. Die Anzahl der Hosts ist ebenfalls um 2 reduziert. Die **kleinste Adresse** im Adressbereich des Subnetzes ist für die Bezeichnung des Netzes (**Netzadresse**) selbst und die **größte Netzwerkadresse** im Subnetz für **Broadcast-Sendungen** reserviert.

Spezielle IP-Adressen:

Anteil **Net-ID**	Anteil **Host-ID**	Beschreibung
nur Nullen	nur Nullen	IP-Adresse des jeweiligen Computers, wird beim Hochfahren verwendet
Net-ID	nur Nullen	Netzwerkadresse, identifiziert ein vollständiges Netzwerk
Net-ID	nur Einsen	Broadcast-Adresse im Netzwerk
127	beliebig	IP-Adresse zum Testen von Netzwerkanwendungen (localhost 127.0.0.1)

Abbildung 11 Spezielle IP-Adressen

3.3 IP-Adressen für private Netzwerke

Innerhalb jeder Adressklasse ist ein Bereich von IP-Adressen festgelegt, die **im Internet nicht verwendet** (geroutet) werden. Im RFC 1918 - *Adress Allocation for Private Internets*, welches die Vergabe von IP-Adressen regelt, werden folgende besondere Adressbereiche ausgewiesen:

- **10.**0.0.0 - **10**.255.255.255: ein vollständiges Class-A Netz
- **172.16.**0.0 - **172.31**.255.255: 16 fortlaufende Class-B Adressen
- **192.168.0**.0 - **192.168.255**.255: 256 fortlaufende Class-C Adressen

Aus diesen Bereichen können zum Aufbau von firmeninternen TCP/IP Netzen (Intranet's) beliebige Adressen ausgewählt werden. Diese Adressen können weltweit beliebig oft verwendet werden.

Die Anbindung eines Intranet ans Internet erfolgt über einen Router, der eine Adressübersetzung durchführt (über **NAT** = **Network Address Translation** oder **PAT** = **Port and Address Translation**). Dabei werden die internen Adressen auf mindestens eine offizielle Adresse übersetzt.

Dies hat den Vorteil, dass einerseits Adressen gespart werden (eine Firma verbraucht nur eine Adresse, obwohl sie viele Rechner im Internet hat) und andererseits beim Wechsel des ISP (Internet Service Provider) nur die NAT-Tabelle am Router zu ändern ist.

3.4 IP V4 - Sub- und Supernetze

Im Zuge der **Adressenverknappung** durch das explosionsartige Wachstum der Internet-Zugänge ist es mittlerweile schwieriger geworden, auch nur ein einzelnes komplettes Class-C Netz zugeteilt zu bekommen.

3.4.1 Subnetting und Supernetting

Unter Subnetting wird allgemein das **Unterteilen eines größeren Netzes in mehrere kleine Netze** verstanden. Die Gründe ein Netz aufzuteilen, können topologischer oder auch organisatorischer Natur sein. Im Gegensatz dazu, werden beim Supernetting mehrere kleinere Netze zu einem größeren Netz zusammengefasst.

3.4.2 Subnetzmaske

Die Subnetzmaske gleicht einer normalen IP-Adresse und **legt den Netz- und Hostanteil einer IP-Adresse fest**. Die von links beginnend, durchgehend gesetzten 1-Bits markieren den Netzanteil, die restlichen durchgehend gesetzten 0-Bits den Hostanteil. **Die bitweise UND-Verknüpfung einer IP-Adresse mit der Subnetzmaske ergibt die Netzadresse.** Ergibt die UND-Verknüpfung für zwei IP-Adressen das gleiche Ergebnis, liegen beide Hosts im selben Netzwerk.

Beispiel:

IP-Adresse:	10101100 00000001 00010001 00001010	bzw. 172.1.17.10
Subnetzmaske:	11111111 11111111 11110000 00000000	bzw. 255.255.240.0
Netzadresse:	10101100 00000001 00010000 00000000	bzw. 172.1.16.0

Mithilfe der Subnetzmaske besteht somit die Möglichkeit, die standardmäßigen Grenzen zwischen dem Netz- und Hostanteil beliebig zu verschieben. Man nennt dieses Verfahren **Classless Inter-Domain Routing**. Eine **CIDR-Adresse** umfasst die 32 Bit IP-Adresse mit durch Schrägstrich getrennter angehängter Anzahl der 1-Bits der Subnetzmaske.

Beispiel Interpretation IP-Adresse:

IP-Adresse:	178.16.35.10	IP Adresse aus Klasse B-Netz
Subnetzmaske:	255.255.255.0	binär: 11111111.11111111.11111111.00000000
CIDR-Adresse:	178.16.35.10/24	24 Bits mit Einsen
Netzadresse:	178.16.35.0	aus Bit-Verknüpfung IP-Adresse & Subnetzmaske
Broadcastadresse:	178.16.35.255	letzte Adresse im Subnetzadressbereich

Zu einer IP-Adresse gehört immer die Angabe der Subnetzmaske und damit die Information wie die Netzklassen segmentiert sind. Die Angabe eine IP-Adresse ohne Subnetzmaske ist wertlos.

Beispiele Subnetting:

1: Das Class-C Netz 192.129.50.0 soll in 2 Subnetze unterteilt werden.

Netzwerk:	**192.129.50.0/24**	
Netzmaske:	11111111 11111111 11111111 00000000	bzw. 255.255.255.0
Netzadresse:	11000000 10000001 00110010 00000000	bzw. 192.129.50.0
Broadcast-Adresse:	11000000 10000001 00110010 11111111	bzw. 192.129.50.255
Erster Host:	11000000 10000001 00110010 00000001	bzw. 192.129.50.1

2: Class-C Netz 192.129.50.0 in 2 Subnetze unterteilt:

Netzwerk 1:	**192.129.50.0/25**	
Subnetzmaske 1:	11111111 11111111 11111111 10000000	bzw. 255.255.255.128
Netzadresse 1:	11000000 10000001 00110010 00000000	bzw. 192.129.50.0
Broadcast im Subnetz 1:	11000000 10000001 00110010 01111111	bzw. 192.129.50.127
Erster Host im Subnetz 1:	11000000 10000001 00110010 00000001	bzw. 192.129.50.1
Netzwerk 2:	**192.129.50.128/25**	
Subnetzmaske 2:	11111111 11111111 11111111 10000000	bzw. 255.255.255.128
Netzadresse 2:	11000000 10000001 00110010 10000000	bzw. 192.129.50.128
Broadcast im Subnetz 2 :	11000000 10000001 00110010 11111111	bzw. 192.129.50.255
Erster Host im Subnetz 2:	11000000 10000001 00110010 10000001	bzw. 192.129.50.129

Was bei Subnetzen allgemein zu beachten ist:

- Mit der Einführung von Subnetzen verringert sich die Anzahl von Rechnern, die jedem Subnetz zugeordnet werden können, da pro Subnetz eine Netz- und Broadcastadresse verloren gehen.
- Alle Rechner eines Subnetzes müssen dieselbe Subnetzmaske verwenden.
- Subnetze sind eine rein lokale Angelegenheit, für die Internet-Welt stellt eine Adresse aus einem Subnetz als ganz normale IP-Adresse dar.

3.5 IP-Paket

Das IP-Protokoll nimmt die Datenpakete der übergeordneten Transportschicht und **zerteilt** sie, falls erforderlich, **in** kleinere **durchnummerierte Einheiten**. Danach bekommt jedes Datenpaket einen IP-Header mit **Sender-** und **Empfänger-IP-Adresse** vorangestellt.

Erhält ein Router ein IP-Paket, das für das Netz in welches das Paket weitergeschickt werden soll zu groß ist, so trennt der Router das Paket in mehrere kleinere Pakete auf, die in die Daten-Frames des betreffenden Subnetzes passen. Haben alle IP-Pakete ihr Ziel erreichen, setzt das IP-Protokoll des Ziel-Hosts die Pakete wieder in der ursprünglichen Reihenfolge zusammen.

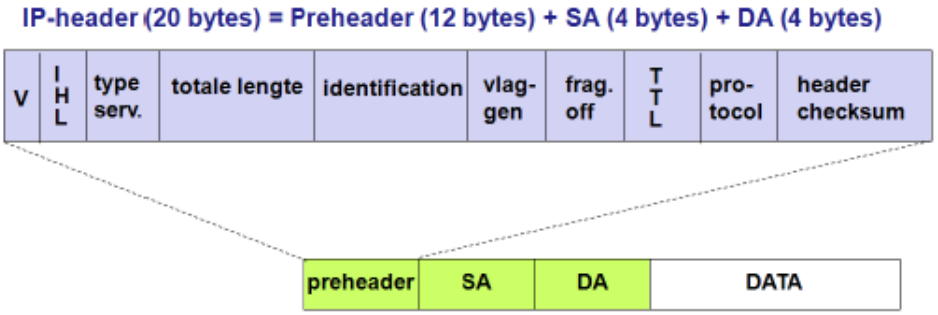


Abbildung 12 IP-Paket

Bedeutung einiger Einträge:

- **Type of Service (ToS)**: zur Priorisierung von IP-Datenpaketen
- **Identifikation**: zur Identifikation der Teilpakete für korrektes Zusammensetzen im Ziel-Host
- **Time to Live (TTL):** Lebensdauer des Pakets. Hat dieses Feld den Wert Null, so wird das Paket verworfen. Jeder Router auf dem Weg des Pakets verringert diesen Wert um eins.
- **Protokoll**: Kennung des nächsthöheren Protokolls: z.B. 01h = ICMP, 06h = TCP, 11h = UDP
- **SA / DA : IP-Adresse des Absenders / Empfängers.**

3.6 Routing in IP-Netzen

Der **Weitertransport** (das "Routing") **von IP-Paketen wird in IP-Netzen von Routern erledigt** und basiert auf der sogenannten **"next - hop" Strategie**. Für die Bearbeitung eines Datenpaketes muss ein Rechner im IP-Netz anhand der Zieladresse über die weitere Verarbeitung folgendermaßen entscheiden:

1. Handelt es sich bei der Zieladresse dabei um die **eigene IP-Adresse**, so wird das Paket intern weiterverarbeitet.
2. Handelt es sich bei der Zieladresse um die **Broadcast-Adresse** des direkt verbundenen Netzwerks, so wird das Paket sowohl, lokal weiterverarbeitet und auch als Broadcast gesendet.

3. Befindet sich die **Zieladresse im direkt verbundenen Netzwerk**, so wird das Paket direkt an den Ziel-Host gesendet. Der Ziel-Host befindet sich im selben Netzwerk, wenn die UND-Verknüpfung der eigenen und der Zieladresse mit der Subnetzmaske dasselbe Ergebnis liefert.
4. Ist die **Zieladresse nicht in einem direkt verbundenen Netz**, so wird das Paket zum weiteren Transport an einen den nächsten Router im direkt verbundenen Netzwerk gesendet. Der Router kann das Paket entweder zustellen, weil das Ziel in einem an ihn direkt verbundenen Netzwerk liegt, oder er reicht es an den nächsten Router weiter.

Um ein Paket zustellen zu können, muss dem Sender die HW-Adresse (= MAC-Adresse) des Ziels bekannt sein. Diese wird vor der Zustellung mittels **ARP-Request** ermittelt. Kommt auf einen ARP-Request keine Antwort, ist das Ziel zurzeit nicht verfügbar ist.

Wenn ein IP-Paket den Router erreicht, entnimmt dieser den Inhalt und **untersucht das IP-Paket**. Der Router benötigt die Information, über welches verbundene Netzwerk er die Nachricht weiterleiten muss. Um das korrekte Netzwerk zu bestimmen, schlägt der Router die Zieladresse des zu routenden Pakets in seiner **Routing-Tabelle** nach.

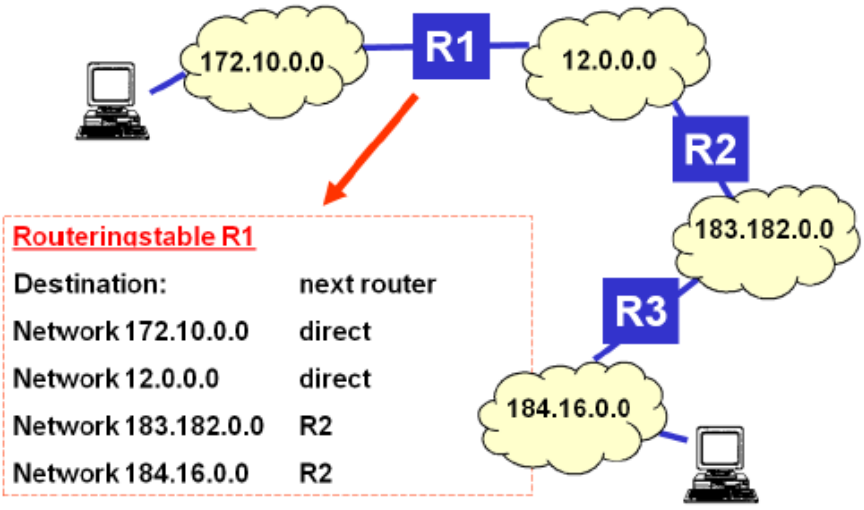


Abbildung 13 Beispiel Routingtabelle

Router geben sich zum Aufbau der Routing-Tabellen gegenseitig ständig Auskunft darüber, welche Netze an ihnen angeschlossen sind und welche Netze ihre Nachbarn kennen. Sie kommunizieren dabei über eigene **Routing-Protokolle**.

Wird die Zieladresse in der Routing-Tabelle gefunden, sendet der Router das IP-Paket mit **ausgetauschten MAC-Adressen** weiter. Es werden die eigene und die MAC-Adresse des nächsten Ziels eingesetzt. Der Inhalt des IP-Paktes bleibt dabei unberührt.

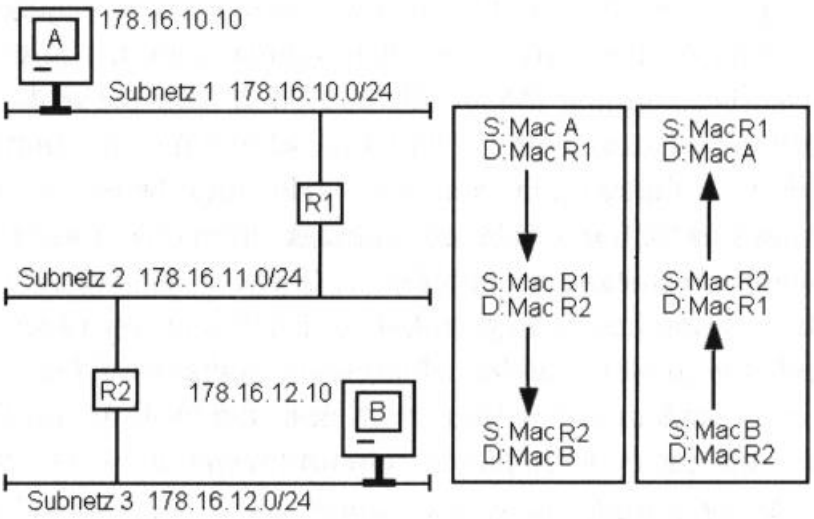


Abbildung 14 MAC-Adressenaustausch

Ein Datenpaket darf normalerweise nur eine, durch den **Time to Live-Wert** (TTL) des Pakets begrenzte Anzahl von Routern passieren, bevor es sein Endziel erreicht. Ist der Wert erreicht, wird das Datenpaket nicht mehr weitergeleitet und verworfen.

Unterschiede zwischen Router und Firewall:

Eine Firewall ist im Prinzip nichts anderes als ein Router, jedoch mit einer invertierten Weiterleitungsphilosophie. Ein **Router ist maximal offen**, bestrebt, hat also das Ziel, so viele Daten als möglich weiter zu transportieren. Hier müssen gewünschte Beschränkungen bewusst konfiguriert werden (sogenannte Access-Control-Lists, ACL). Unerwünschter Verkehr muss also bewusst durch Konfiguration verboten werden. Eine **Firewall ist maximal geschlossen** und verhält sich damit genau anders herum. Hier ist jeglicher Verkehr von vornherein verboten. Will man bestimmte Daten weiterleiten, muss dies explizit durch Konfiguration erlaubt werden.

3.7 Internet Control Message Protocol (ICMP)

Die TCP/IP-Protokoll-Suite enthält das ICMP zum **Test der Netzfunktionalität**, **Versand von Fehlermeldungen** und für weiter **Informationsdienste**.

Mit dem ICMP wird gemeldet, wenn eine bestimmte Netzwerkeinrichtung nicht verfügbar ist, oder ein bestimmter Host oder Router nicht erreichbar ist. Die Diagnoseprogramme ***ping*** und ***tracert*** (*traceroute* unter Linux) basieren auf dem ICMP.

Das ICMP verwendet IP-Pakete zum Versand von Nachrichten. Eine ICMP-Fehlermeldung wird stets als Antwort auf ein bestimmtes IP-Paket verarbeitet und an dessen Quelle zurückgesendet.

Konsolenkommando *ping*:

Eines der wichtigsten Diagnosewerkzeuge ist der *Ping*-Befehl. Dieser sendet, über ICMP IP-Pakete an einen anderen Teilnehmer, um zu überprüfen, ob dieser Host über das Netzwerk erreichbar ist.

```
c:\sysinternals>ping www.google.de

Ping wird ausgeführt für www.google.de [172.217.19.99] mit 32 Bytes Daten:
Antwort von 172.217.19.99: Bytes=32 Zeit=44ms TTL=49
Antwort von 172.217.19.99: Bytes=32 Zeit=42ms TTL=49
Antwort von 172.217.19.99: Bytes=32 Zeit=43ms TTL=49
Antwort von 172.217.19.99: Bytes=32 Zeit=46ms TTL=49

Ping-Statistik für 172.217.19.99:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 42ms, Maximum = 46ms, Mittelwert = 43ms
```

Konsolenkommando *tracert*:

Der Befehl ***tracert*** zeigt den Weg und alle Zwischenstationen eines Datenpakets zwischen zwei Hosts an. Tracert kann sowohl mit IP- als auch Host-Namen genutzt werden.

```
C:\Windows\system32>tracert www.google.de

Routenverfolgung zu www.google.de [216.58.211.3]
über maximal 30 Hops:

  1     6 ms     1 ms     1 ms  192.168.2.1
  2     2 ms     1 ms     1 ms  fritz.fon.box [192.168.178.1]
  3    22 ms    20 ms    19 ms  80.66.48.1
  4    23 ms    22 ms    21 ms  80.66.62.49
  5    28 ms    22 ms    23 ms  80.66.62.162
  6    24 ms    21 ms    21 ms  80.66.58.158
  7    27 ms    22 ms    27 ms  80.66.58.234
  8    33 ms    24 ms    25 ms  80.66.62.150
  9    45 ms    46 ms    38 ms  de-cix10.net.google.com [80.81.192.108]
 10    42 ms    42 ms    40 ms  216.239.47.18
 11    47 ms    41 ms    42 ms  209.85.243.131
 12    50 ms    47 ms    51 ms  216.239.40.7
 13    48 ms    48 ms    48 ms  209.85.142.14
 14    48 ms    50 ms    46 ms  216.239.47.189
 15    74 ms    47 ms    46 ms  muc03s13-in-f3.1e100.net [216.58.211.3]
```

tracert sendet Testpakete mit jeweils um 1 erhöhten **TTL**-Werte beginnend mit dem Wert 1. Jeder Router verringert den Wert um 1, verwirft die Nachricht bei Erreichen von TTL= 0 und sendet in diesem Fall die ICMP-Fehlermeldung *Time Exceeded* zurück. Auf diese Weise kann die IP-Adresse jedes Routers auf dem Weg zum letzten Host ermittelt werden.

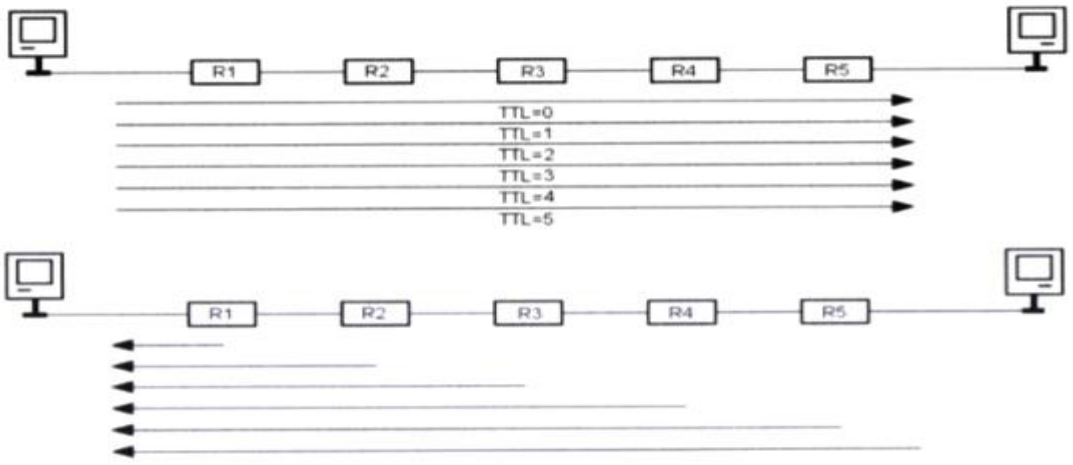


Abbildung 15 Testpakete von tracert

3.8 Domain Name System (DNS)

Das DNS wird zur **Umsetzung von Domainnamen in IP-Adressen** benutzt. Dies ist vergleichbar mit einem Telefonbuch, das die Namen der Teilnehmer in ihre Telefonnummer auflöst. Mit dem DNS ist auch eine **umgekehrte Auflösung von IP-Adressen in Namen** möglich.

Das DNS ist ein weltweit auf tausenden von Servern verteilter **hierarchischer Verzeichnisdienst**, der den **Namensraum des Internets** verwaltet. Dieser Namensraum ist in **Zonen** unterteilt, für die jeweils **unabhängige Administratoren** zuständig sind. Für lokale Anforderungen, etwa innerhalb eines Firmennetzes, ist es auch möglich, ein vom Internet unabhängiges DNS zu betreiben.

Das DNS zeichnet sich aus durch:

- **dezentrale Verwaltung**
- **hierarchische Strukturierung des Namensraums** in Baumform
- **Eindeutigkeit der Namen**
- **Einfache Erweiterbarkeit**

3.8.1 Domain-Namensraum

Ein Eintrag im DNS besteht aus mehreren Namen**,** auch **Labels** genannt, die **durch „.“ getrennt** sind. Diese Labels beschreiben den Pfad durch den DNS-Baum. Ein Domainname wird immer von **rechts nach links** aufgelöst, das heißt, je weiter rechts ein Label steht umso höher steht er im Baum. Eine Domain umfasst den gesamten untergeordneten DNS-Namensraum (z.B. alle Namen unterhalb ac.at.). Eine Zone eine oder mehrere Ebenen innerhalb einer Domäne.

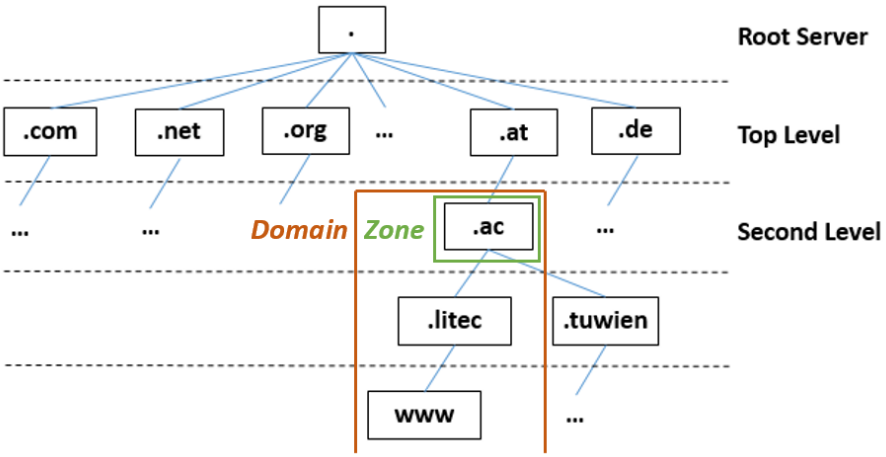


Abbildung 16 Hierarchische Struktur des DNS

3.8.2 Nameserver

Nameserver sind **Programme**, die Anfragen zum Domain-Namensraum beantworten. Für jede Zone muss ein **primärer Nameserver** und aus Redundanz- und Lastverteilungsgründen ein oder mehrere **sekundäre Nameserver** vorhanden sein. Der primäre Nameserver hält die **gesicherten Daten** einer Zone. Jeder sekundäre Nameserver beziehen seine Daten vom primären Nameserver. Den Datentransfer vom primären zum sekundären Server wird als **Zonentransfer** bezeichnet**.**

Zusätzlich unterscheidet man zwischen **autoritativen** und **nicht-autoritativen** Nameservern. **Ein autoritativer Nameserver ist verantwortlich für eine Zone**. Seine Informationen werden deshalb als **gesichert** angesehen. Für jede Zone existiert mindestens ein autoritativer Server, der primäre Nameserver**.** Ein **nicht-autoritativer Nameserver bezieht seine Informationen** über eine Zone von anderen Nameservern sozusagen aus zweiter oder dritter Hand. Seine Informationen werden als **nicht gesichert** angesehen.

Da sich DNS-Daten normalerweise nur sehr selten ändern, speichern nicht-autoritative Nameserver die einmal angefragten Informationen lokal ab, damit diese bei einer erneuten Anfrage schneller vorliegen. Jeder dieser Einträge besitzt ein eigenes **Verfallsdatum** (**TTL** Time to Live), nach dessen Ablauf der Eintrag aus dem Cache gelöscht wird.

3.8.3 Resolver

Möchte ein Programm, z.B. ein Webbrowser, einen Hostnamen auflösen um die entsprechende Web-Adresse zu kontaktieren, so schickt er eine Anfrage an den **Resolver**. Dies kann ein **Teil des Betriebssystems** oder auch ein eigenes Programm sein. Der Resolver durchsucht zuerst seinen lokalen Cache, ob er die Anfrage beantworten kann. Falls dies fehlschlägt, kontaktiert einen **externen Nameserver**.

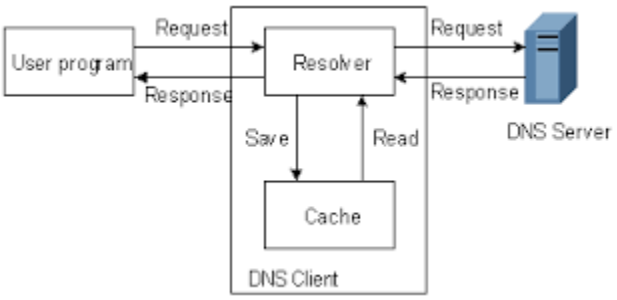


Abbildung 17 Funktionsweise Resolver

Beispiel **Forward Lookup** (= IP Adresse aus DNS-Namen bestimmen) und **Reverse Lookup** (=DNS-Name aus IP-Adresse):

Konsolenkommando *nslookup:*

Mit *nslookup* wird eine Anfrage an einen Nameserver geschickt.

```
C:\>nslookup www.tuwien.ac.at
Server:     homerouter.cpe
Address:    fe80::be3d:85ff:fe07:8f0f

Nicht autorisierende Antwort:
Name:       info.zserv.tuwien.ac.at
Addresses:  2001:629:1006:80::76
            128.130.35.76
```

```
C:\>nslookup 128.130.35.76
Server:     homerouter.cpe
Address:    fe80::be3d:85ff:fe07:8f0f

Name:       info.zserv.tuwien.ac.at
Address:    128.130.35.76
```

3.9 Dynamic Host Configuration Protocol (DHCP)

DHCP ermöglicht die **vollautomatische Einbindung eines neuen Hosts** in ein bestehendes Netzwerk **ohne weitere Konfiguration**. Am Host muss im Normalfall lediglich der **automatische Bezug der IP-Adresse** eingestellt sein. Beim Start des Hosts kann dieser die **IP-Adresse**, **Netzmaske**, **Gateway**, **DNS-Server** vom DHCP-Server automatisch beziehen.

Das DHCP wurde im Hinblick auf zwei **Einsatzszenarien** entwickelt:

- Große Netzwerke mit **häufig wechselnder Topologie**
- Bedarf einer Netzwerkverbindung **ohne aufwändige Netzwerkkonfiguration**

Damit die vom DHCP-Server bezogenen IP-Adressen nach dem Abschalten von Klienten wieder frei werden, besitzen die IP-Adressen ein **Verfallsdatum**, das vom DHCP-Server vorgegeben wird. Ist es abgelaufen, muss der Klient eine Verlängerung beantragen. Kommt dieser Antrag nicht, legt der DHCP-Server die Adresse wieder in den Pool für die nächste Anfrage bereit.

Die meisten DHCP-Server bieten die Möglichkeit einer **MAC-Adressbindung**. Durch eine feste Zuordnung von bekannten MAC-Adressen zu reservierten IP-Adressen im kann man bewirken, dass nur bekannte Rechner IP-Adressen zugeteilt bekommen, oder dass ein Rechner trotz dynamischer Zuteilung immer dieselbe Adresse bekommt.

Die **MAC-Adressenbindung als Sicherheitsfunktion** hat aber Lücken. Bei vielen Netzwerkadaptern ist es möglich, die MAC-Adresse selbst zu konfigurieren. Dadurch ist die eindeutige Identität eines Netzwerkgerätes über die MAC-Adresse nicht mehr gegeben.

Address Reservation

ID	MAC Address	Reserved IP Address
1	18-5E-0F-30-B0-56	192.168.2.100
2	B8-27-EB-0E-26-B9	192.168.2.113

Abbildung 18 DHCP Address Reservation

3.10 Network Address Translation (NAT)

Die Adressen im Internet werden langsam knapp. Große Adressklassen stehen nicht mehr zur Verfügung, daher mussten Methoden gefunden werden, um die **Adressen** weiter zu „**strecken**“. Hinter den folgenden beschriebenen Verfahren stehen aber auch **Security-Überlegungen**.

Das erste Verfahren war NAT, die *Network Address Translation*. Ein Router oder eine Firewall ist in der Lage, die **privaten IP-Adressen** in den abgesendeten Datenpaketen **gegen öffentliche Adressen auszutauschen**. Führt er Buch über seine Änderungen, können die Änderungen in den Antwortpaketen wieder rückgängig gemacht werden.

So ist es möglich, dass viele Rechner im privaten Netz einer Firma betrieben werden können und nur ausgewählte Rechner Zugang zum Internet erhalten. Ein weiterer Vorteil neben dem Einsparen von Adressen ist die Security. Nicht jeder Rechner ist von außen erreichbar. Die interne Netzwerkstruktur bleibt verborgen.

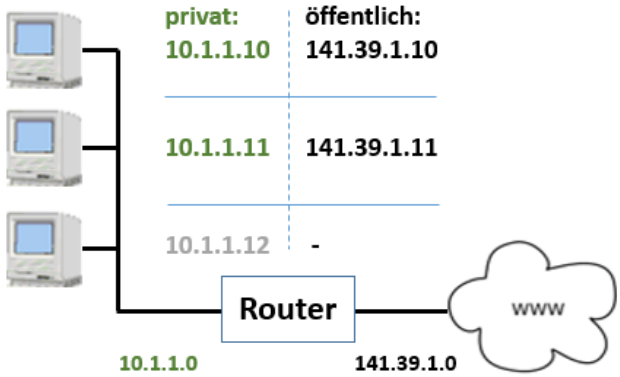


Abbildung 19 NAT

3.11 Port and Address Translation (PAT)

Die Weiterentwicklung von NAT ist PAT, Port oder **IP-Masquerading** genannt. Hier werden **mehrere private IP-Adressen in eine einzelne öffentliche IP-Adresse** übersetzt. Dies ist möglich, wenn zusätzlich zu den Adressen auch die **Portnummern umgeschrieben** werden.

In jedem Datenpaket sind die Source-Adresse und der Source-Port genauso wie die Destination-Adresse und der Destination-Port enthalten. Ersetzt nun der Router in jedem Datenpaket nach außen die Angaben zu den Adressen und zum **Source-Port des Senders** und führt darüber genau Buch, kann er die Antworten auf dem Rückweg wieder zu den dedizierten Rechnern zurückleiten und die ursprünglichen Adressen und Ports wieder eintragen.

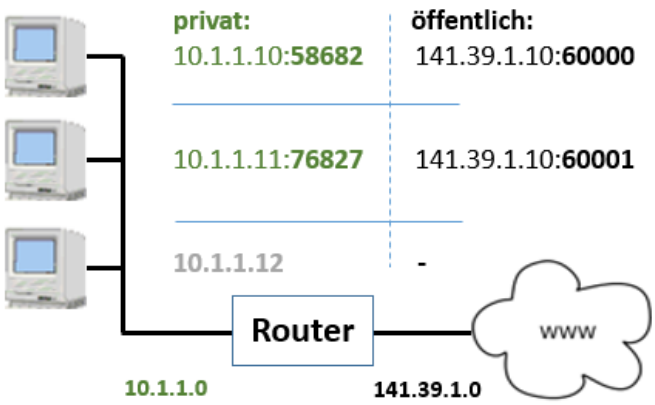


Abbildung 20 PAT

3.12 IP-V6

Grund für die "neue Generation" von IP ist die rapide Verknappung der freien Internet-Adressen und die zunehmende Unhandlichkeit der Routing Tabellen. Der Adressenvorrat von IP V4 ist praktisch erschöpft.

IP V6-Adresse:

- Eine IPV6-Adresse ist **128 Bit** lang. Damit gibt es etwa 3,4 x 10^{38} IPV6- Adressen - für jeden Quadratmeter Erdoberfläche könnten 6,7 x 10^{23} Adressen bereitgestellt werden.
- IPV6-Adressen werden nicht mehr dezimal sondern **hexadezimal in acht Blöcken** mit Doppelpunkten getrennt geschrieben. z.B.: 243f:6a88:85a3:0000:1319:8a2e:0370:7344
- Führenden Nullen können weggelassen werden. Aufeinanderfolgende **Blöcke von Nullen** können einmal innerhalb der gesamten Adresse durch **::** ersetzt werden.

  z.B.: **243f:6a88:85a3::1319:8a2e:370:7344** statt 243f:6a88:85a3:0000:1319:8a2e:0370:7344
- In Internet Adressen wird die IP V6-Adresse in **eckige Klammern** eingeschlossen.
  z.B.: http://**[**243f:6a88:85a3:08d3:1319:8a2e:0370:7344**]**:80/

Die ersten **64 Bit** der IP V6-Adresse bilden den sogenannten **Network-Prefix** und dienen der **Netzadressierung**, die letzten **64 Bit** bilden bis auf Sonderfälle einen für die **Netzwerkschnittstelle eindeutigen Interface-Identifier**. Mit IP V6 werden nicht mehr ein Host, sondern es werden seine Interfaces adressiert.

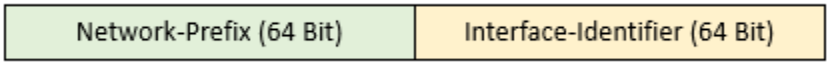


Abbildung 21 Zusammensetzung IPV6 Adresse

IP V6-Netzwerke werden in der **Classless Inter-Domain Routing** (CIDR) Notation aufgeschrieben. Dazu werden die Netzadresse und die Länge des Präfixes in Bits, durch einen Schrägstrich getrennt.

Zum Beispiel steht 2001:0db8:1234::/48 für das Netzwerk mit den Adressen
`2001:0db8:1234:0000:0000:0000:0000:0000 - 2001:0db8:1234:ffff:ffff:ffff:ffff:ffff`

Typischerweise bekommt ein **Internetprovider** die ersten **32 Bit als Netz** zugewiesen. Dieser Bereich wird vom Provider in weitere Subnetze aufgeteilt. Die Länge der Zuteilung an Endkunden wird dabei dem Provider überlassen. Vorgeschrieben ist die minimale Zuteilung eines /64-Netzes. Als kleinste Einheit bekommt der **Kunde** also **2^{64} Adressen**!

Parallelbetrieb IPV4 - IPV6:

Um einen einfachen Übergang von IP V4- zu IP V6-Kommunikation im Internet zu ermöglichen, wurden verschiedene Mechanismen entwickelt. IP V6 wird dabei in der Regel hinzugeschaltet, ohne IP V4 abzuschalten. Grundlegend werden folgende drei Mechanismen unterschieden:

- **Parallelbetrieb (Dual Stack):**

Bei diesem Verfahren wird allen beteiligten Schnittstellen neben der IP V4-Adresse **zusätzlich mindestens eine IPV6-Adresse** zugewiesen. Die Rechner können dann über beide Protokolle unabhängig kommunizieren.

- **Tunnelmechanismen:**

Um Netze zu überbrücken, die IPV6 nicht weiterleiten, gibt es unterschiedliche Tunnel-mechanismen. Im Tunnelbetrieb werden IPV6-Pakete z.B**. in die Nutzdaten von IPV4-Paketen gepackt** und bis zu einer IPV6-Tunnelgegenstelle als IPV4-Paket übertragen. Dort werden die IPV6-Pakete herausgelöst und zum Ziel via IPV6-Routing übertragen. Der Rückweg funktioniert analog.

- **Übersetzungsverfahren:**

Kann auf einem Gerät IP V6 nicht aktiviert werden oder stehen nicht mehr genügend IP V4-Adressen zur Verfügung, können Verfahren wie *Network Address Translation* nötig werden, um zwischen beiden Protokollen zu übersetzen.

4 Transportschicht, Layer 4

Die Transportschicht ist für die **Ende-zu-Ende-Übertragung von Daten zwischen Anwendungen** (Prozessen) zuständig, unabhängig von der Art der Daten und von der Art und Weise, wie die Daten übermittelt werden.

Zwei Protokolle werden hauptsächlich in dieser Schicht verwendet: das **Transmission Control Protocol (TCP)** und das **User Datagram Protocol (UDP)**. TCP ist ein verbindungsorientiertes, zuverlässiges Protokoll. UDP hingegen stellt nur sicher, dass die Pakete an die richtige Anwendung geschickt werden.

TCP ist neben IP der Namensgeber für das gesamte Modell (TCP/IP-Modell), weil es das am häufigsten verwendete Protokoll in der Transportschicht ist.

4.1 Transmission Control Protocol (TCP)

Das TCP-Protokoll ist verantwortlich dafür, **Informationen korrekt** über ein oder mehrere Netzwerke zu übertragen. Die Datenpakete eines Datenstromes werden, unterwegs fragmentiert, sie erreichen das Ziel unter Umständen über verschiedene Wege und in der falschen Reihenfolge. TCP **setzt die Daten wieder richtig zusammensetzt**. Dabei kontrolliert TCP, ob alle Pakete eingetroffen sind, und fordert verlorene vom Sender nach.

TCP ist ein **verbindungsorientiertes** Protokoll. Es wird eine **logische Verbindung** aufgebaut, verwendet und danach wieder beendet. Aus der Sicht des TCP ist das gesamte Internet ein Kommunikationssystem, das Nachrichten senden und empfangen kann, ohne deren Inhalt zu verändern oder zu interpretieren.

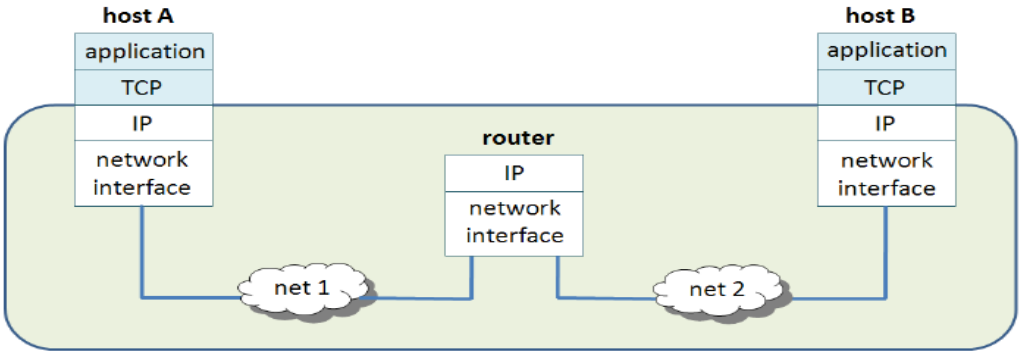


Abbildung 22TCP als Ende-zu-Ende Protokoll

Three-Way-Handshake:
Bevor Daten ausgetauscht werden können, muss die Verbindung mittels des TCP-Handshakes aufgebaut werden. Um zu gewährleisten, dass Verbindungen auf zuverlässige Weise aufgebaut und wieder beendet werden, verwendet das TCP Protokoll ein Three-Way-Handshake, bei dem drei Nachrichten ausgetauscht werden.

- Client schickt ein **SYN-Paket an den Server**
- Server antwortet mit einem **SYN/ACK-Paket an den Client**
- Client schließt den Verbindungsaufbau mit einem **ACK-Paket an den Server** ab

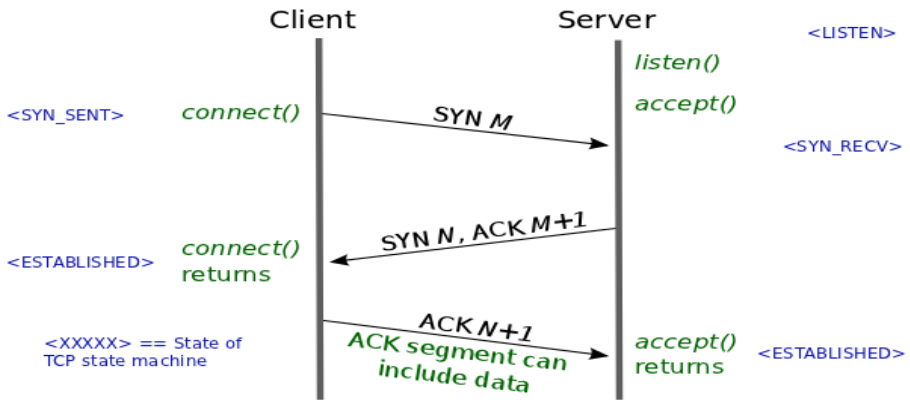


Abbildung 23 TCP Three-Way-Handshake

1.1.1 Das TCP-Segment

Die zu versendenden Informationen werden von der Anwendungsschicht an die Transportschicht weitergegeben. Die Transportschicht packt die Informationen in das Datenfeld und fügt dann den TCP-Header hinzu. Dieses Paket wird dann an die IP-Schicht übergeben.

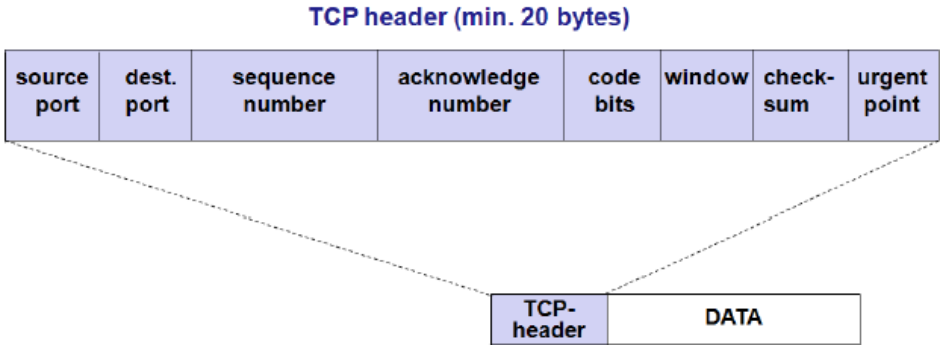


Abbildung 24 TCP-Header

Bedeutung einiger Einträge:

- **Source- und Destination-Port**: Port-Nummern der sendenden und empfangenden Applikation.
- **Sequence-Number**: Nummer des Datenpakets im Datenstrom.
- **Acknowledge-Number**: Jedes Paket wird vom Empfänger bestätigt. Diese Nummer gibt an welches Paket richtig angekommen ist.

4.2 User Datagram Protocol (UDP)

Mit UDP können Anwendungen IP-Pakete senden, ohne eine Verbindung aufbauen zu müssen. UDP ist ein **verbindungsloses Protokoll**, bei dem die Zustellung der Daten nicht gesichert ist.

- es gibt **keine Empfangsquittierung**
- die **Reihenfolge** der ankommenden Pakete kann sich ändern.
- es können Pakete **mehrfach oder gar nicht ankommen**.

UDP erstellt ein IP-Paket mit einem sehr kleinen Rahmen (nur 8 Byte), so dass UDP nicht viel mehr ist als eine Art **IP auf Anwenderebene**.

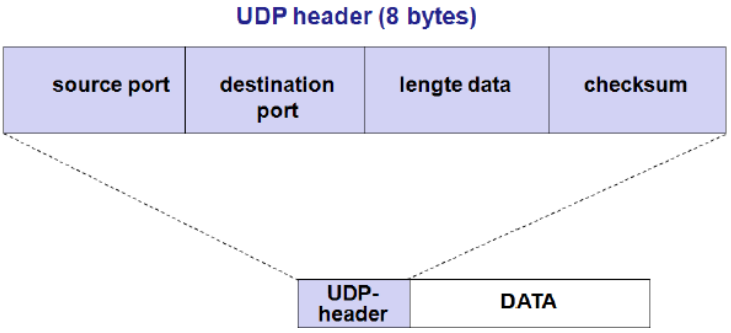


Abbildung 25 UDP-Segment

UDP eignet sich daher auch für **Übertragung von Echtzeitdaten** (Telefon, Videokonferenz, Real Audio). Die Geschwindigkeit steht hier im Vordergrund gegenüber der Übertragungssicherheit, denn die genannten Anwendungen verkraften erfahrungsgemäß das Ausbleiben einzelner Datenblöcke. Die **Absicherung liegt hier alleine bei der Anwendung** und ist nicht im Protokoll verankert.

4.3 Ports

Auf einem Host können gleichzeitig mehrere Clients und Server aktiv sein. Dabei ist es wichtig, dass **jede Anwendung und jeder Dienst eindeutig identifizierbar** ist. Zu diesem Zweck gibt das TCP und UDP Protokoll jeder Anwendung und jedem Dienst eine **eindeutige Port-Nummer** als Kennung.

Will ein Client eine Verbindung mit einem Server aufnehmen, sucht er sich **dynamisch einen freien** Port als **Source-Port** und sendet an den **Destination-Port** des Servers. Dieser ist in der Regel ein **Well Known Port**. Ist der Service dort aktiv, werden ihm anhand der Portnummer die Daten übergeben. Die Antwort erfolgt an den Source-Port des Absenders.

Es gibt drei Bereiche für Portnummern:

- **1 - 1023**: „wohlbekannte“ Ports (**Well Known Ports**) für Standardanwendungen.
- 1024 – 49151: Registrierte Ports, die von der IANA nicht kontrolliert, aber registriert werden.
- 49152 – 65535: Ports werden vom Betriebssystem dynamisch an Clientprogramme vergeben.

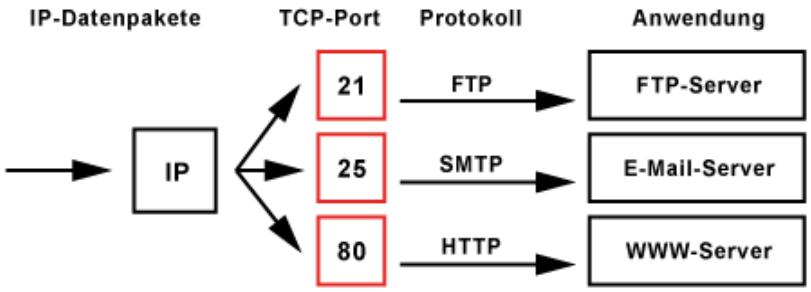


Abbildung 26 Well Known Ports fpr FTP, SMTP, HTTP

4.4 Endpunkt und Socket

Eine **Kombination** aus **IP-Adresse** und **Portnummer** nennt man **Endpunkt**. Ein Endpunkt beschreibt, über welche **logische Adresse** eine Anwendung in einem Netzwerk erreichbar ist. Die Portnummer wird dabei an die IP-Adresse direkt angehängt, getrennt durch einen Doppelpunkt, so zum Beispiel 192.168.17.4:80.

Der Begriff **Socket** ist ein reiner **Software-Begriff**. Ein Socket ist ein Programmier-Interface (Adressen-Datenstruktur) für Kommunikationsprotokolle.

Ein Socket wird durch folgende drei Angaben spezifiziert:

- **Protokoll** (TCP oder UDP)
- **IP-Adresse** (lokale Adresse)
- **Portnummer** (lokaler Prozess oder Anwendung)

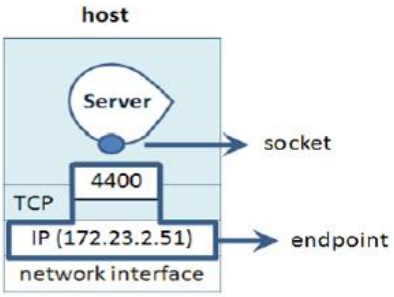


Abbildung 27 Endpunkt und Socket

4.4.1 Socket Programmierung

Voraussetzungen für Verbindungsaufbau:

- Server-Prozess muss laufen
- Server muss einen Eingangs-Socket angelegt haben, der Client-Anfragen entgegen nimmt

Verbindungsaufbau im Client:

- Anlegen eines eigenen Client-TCP-Sockets
- Verbinden mit der IP-Adresse und dem Port des Serverprozesses

Verbindungsaufbau im Server:

Wird der Serverprozess vom Client kontaktiert, dann erzeugt er einen neuen Socket, um mit diesem zu kommunizieren. So kann der Server mit mehreren Clients kommunizieren.

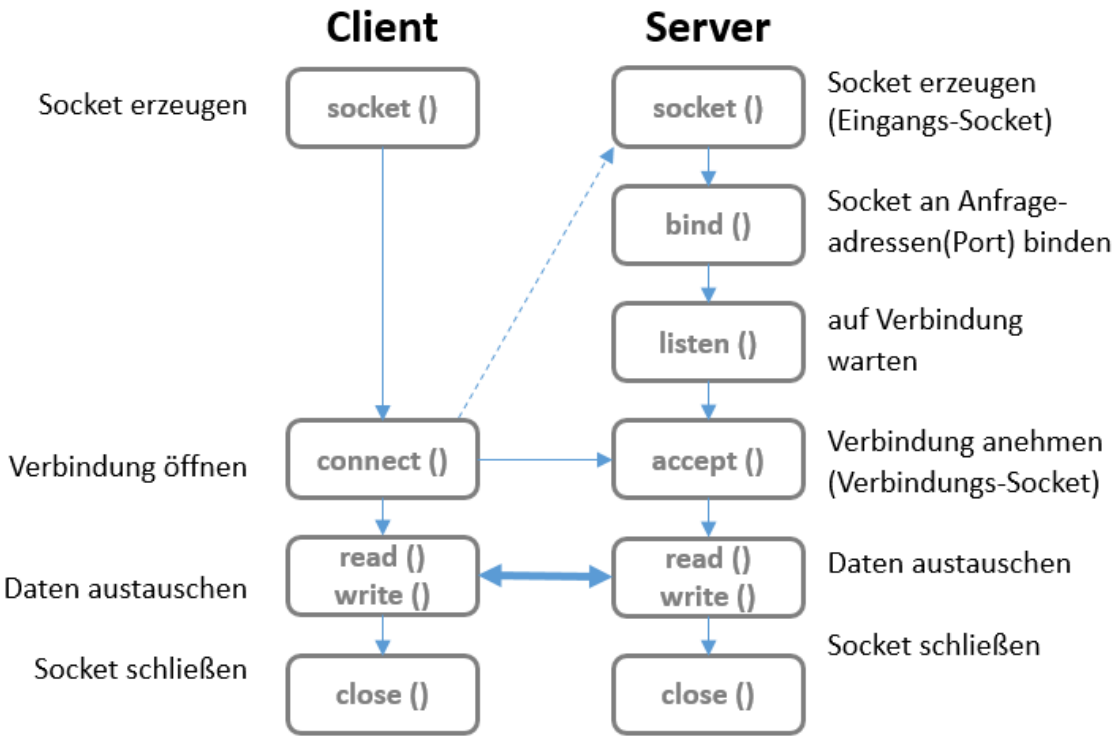


Abbildung 28 TCP-Verbindungsaufbau

5 VLAN - virtuelle lokale Netzwerke

Die Abkürzung VLAN steht für **virtuelles LAN**, also ein Netzwerk mit allen Eigenschaften eines gewöhnlichen LAN, **jedoch ohne räumliche Einschränkungen**. Während die Stationen eines LAN nicht beliebig weit auseinander liegen können, ermöglicht es ein VLAN, weiter entfernte Knoten zu einem virtuellen lokalen Netzwerk zu verbinden.

Grundsätzlicher Aufbau von VLANs:

- Unterteilung eines physikalischen Netzwerks
- auf Layer-2 (Switch-Ebene)
- in mehrere Schicht-2-Netzwerke
    - d.h. Broadcast-Domänen auch über mehrere Switches
    - die voneinander unabhängig sind
    - durch Einsatz von speziellen LAN-Switches

Kommunikation in und zwischen VLANs:

- In VLANs wie in einem normalen LANs
- Zwischen VLANs nur über Routing-Funktionen
    - in Form eines Routers

5.1 Vorteile von VLANs

VLANs vereinen die Vorteile von Switches und Routern. Router sind gegenüber Switches langsamer und teuer und es entstehen viele Subnetze. Der Adressraum wird schnell zu klein. Ein VLAN kann ohne physikalische Veränderungen des Netzwerks eingerichtet werden. Im Idealfall kann die Konfiguration über Software erfolgen.

- **Ortsunabhängige Segmentierung** von Broadcast-Domänen - Broadcast nicht auf allen Ports
- **Performance-Vorteile** von Switches gegenüber Routern - Bearbeitung auf Schicht 2 statt 3
- **Einfache Administration** von Switches - leichtes hinzufügen/entfernen von Stationen
- **Geringere Kosten** von Switches - im Vergleich zu Routern

5.2 Realisierung von VLANs

Im Switch wird festgelegt, welcher Anschluss welchem Segment zugeordnet ist. Rechner können nur mit Rechnern kommunizieren, die im selben Segment (VLAN) angesiedelt sind. Ein Verkehr zwischen den Segmenten ist ohne zusätzlichen Router nicht möglich.

Zur Unterscheidung der Pakete der VLAN-Segmente wurde im Ethernet-Frame eine zusätzliche **Segmentkennung** (VLAN-Tag) festgelegt. Man sagt, ein Paket ist **tagged**, wenn es eine Segment-kennung enthält. Das Paket wird dadurch nach dem Standard 802.1q für VLANs statt 1518 Byte 1522 Byte lang.

Ältere Switches können eventuell die erweiterten Pakte nicht erkennen und verwerfen sie, da sie für Standard-Ethernet zu groß sind. Anhand des VLAN-Tags erkennen die Switches, zu welchem Segment welches Paket gehört.

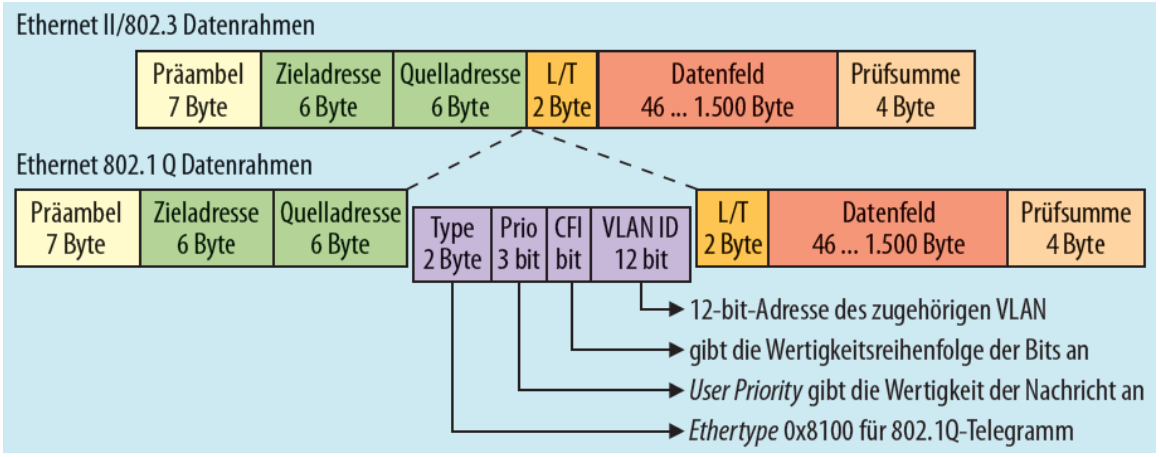


Abbildung 29 VLAN-Tag im Ethernet Frame

Zur Konfiguration Anschluss-Ports eines Switches gibt es zwei mögliche Alternativen. Bei beiden Methoden muss konfiguriert werden, welche Segmente (VLANs) überhaupt erlaubt sind.

MAC-Adressen basierende VLANs:

Die Switches bekommen eine Tabelle mit MAC-Adressen und den ihnen zugehörigen Kommunikationsbereichen. In den MAC-Adresstabellen sehen sie, wo welcher Rechner angeschlossen ist, und leiten ihm die Pakete für sein Segment zu.

Port basierende VLANs:

Die andere Methode ist, dass der Administrator festlegt, welcher Port zu welchem Segment gehört. Letzteres wird zum **Standard**, da es **statisch** ist und nicht erfordert, dass bei einem Ersatz der Hardware nachkonfiguriert werden muss.

5.3 Trunks

VLAN-Tags sind nur bei der Verbindung Switch-zu-Switch im Paket enthalten. Kommen sie am Zielport an, entfernt der Switch die Tags, bevor er sie an den Rechner schickt. Sendet ein Rechner, der in einem bestimmten VLAN ist, fügt der Switch den Tag ein, bevor er das Paket an den nächsten Switch verschickt. Die Rechner bekommen von diesem Vorgang also überhaupt nichts mit. Eine **Verbindung** zwischen Switches, **die getaggte Pakete transportiert**, nennt man einen **Trunk**.

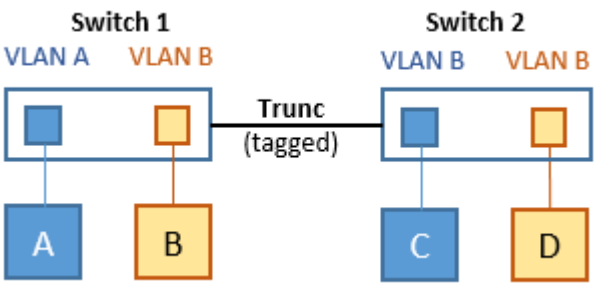


Abbildung 30 Trunc

5.4 Verkehr zwischen VLANs

Ein Switch ist ein Gerät auf Schicht 2, er kann nicht routen. Die Trennung in verschiedene Segmente musste aber einen Layer höher angesetzt werden. Diese Aufgabe kann daher nur ein Router durchführen.
Der Router sieht also virtuell eine Verbindung zu zwei Switches, die in verschiedenen Subnetzen angeordnet sind. Er routet die Daten ganz normal. Der Switch sieht nichts vom Routing, das ist Schicht 3. Er schickt ein Frame zu einem Port heraus, das er **ungetaggt an den Router abgibt**.

Der Router nimmt das Frame auf, schaut nach der IP-Adresse des Zieles und routet das Paket an ein anderes Interface. Das Frame kommt nun in einen anderen Port des Switches herein. Dieser versieht es mit dem Tag des dort konfigurierten VLANs und leitet es weiter. **Am Ziel werden alle Tags entfernt**, und die Daten werden an das Endgerät ausgegeben.

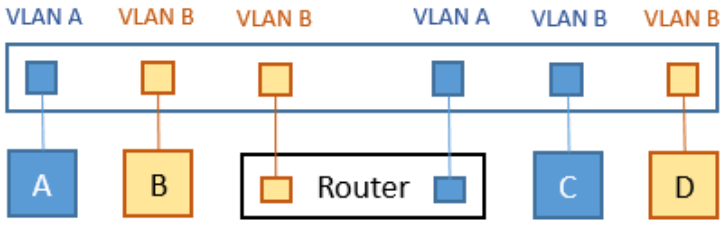


Abbildung 31 Routing in VLANs – Router belegt je Netz einen Port

Moderne Router sind in der Lage, das **Tagging zu verstehen**. Man kann sie also ***einarmig*** an den Switch anschließen. Dies spart teure Router-Interfaces. Auch diese Verbindungen nennt man Trunk-Links. Die Router empfangen alle Pakete *tagged* wie ein Switch und erledigen intern das Routing.

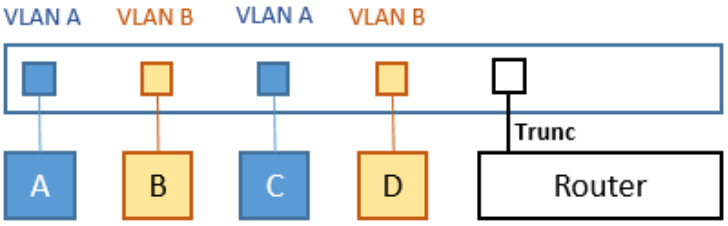


Abbildung 32 Routing in VLANs – Router über Trunc-Link

6 Anwendungsschicht, Layer 5..7

Die Protokolle der Schichten 5-7 des OSI-Modells sind **anwendungsorientiert** und jeweils mit speziellen Diensten gekoppelt. Meistens sind in einer Anwendung alle drei Schichten vereint.

6.1 Dienste und Protokolle im Internet

Im Internet gibt es Computer, die Dienste anbieten, sogenannte Server und Computer, die Dienste in Anspruch nehmen, sogenannte Clients. Eine erfolgreiche Kommunikation der Clients mit den Servern erfordert für jeden Dienst ein Protokoll. Die Server bieten ihre Dienste unter bestimmten *Well Known* Portnummern an.

Protokolle der Schichten 5-7 sind:

- **http** - Hypertext Transfer Protocol
  Übertragung von beliebigen **Daten** über IP-Netzwerke (vorrangig Webseiten)
- **ftp** - File Transfer Protocol
  Übertragung von **Dateien** über IP-Netzwerke.
- **smtp** - Simple Mail Transfer Protocol
  Austausch von **E-Mails** über IP-Netzwerke.

Dienste:

- **dns** - Domain Name System
  Beantwortung von Anfragen zur **Namensauflösung** (basierend auf tcpip)
- **www** – World Wide Web
  Bereitstellung von **Webseiten** mithilfe eines Webbrowser von Webservern (basierend auf http)

Viele Protokolle sind reine ASCII-Protokolle, d.h. die Daten gehen im Klartext über das Netz.
Da einige Dienste das Übertragen von Passwörtern erfordern, ist hier Vorsicht angebracht.
Wichtige Daten sollten nur verschlüsselt übertragen werden.

6.2 http: Hypertext Transfer Protocol

Das Hypertext Transfer Protocol ist das wichtigste und am häufigsten verwendete auf TCP/IP basierende Protokoll im World Wide Web. Allgemein kann man http als ein Protokoll beschreiben, mit dem man **Daten**, hauptsächlich Webseiten, **aus dem www** in einem Webbrowser laden kann. http nicht nur auf die Übertragung von Textseiten beschränkt, sondern kann auch zum Austausch von Grafiken, Videos, und anderen beliebigen Dateien eingesetzt werden.

6.2.1 Geschichte

Die Geschichte des **http** begann lange nach dem Start des Internet im Jahr 1989. Die Idee war, dass jeder Computer der innerhalb des **www** Daten anbieten konnte und dass man von diesem

dann auch bestimmte Daten beziehen konnte.

Tim Berners-Lee und sein Team am CERN, dem europäischen Kernforschungszentrum in der Schweiz, begannen mit der Entwicklung von **http** zusammen mit den Konzepten **url** und **html**, womit die Grundlagen des World Wide Web geschaffen wurden. Erste Ergebnisse dieser Bemühungen war 1991 die Version http/0.9.

Aktuelle http Versionen:

http/1.1 - Spezifikation von 1999-2007

http/2.0 - 2015, vorrangig Maßnahmen zur Beschleunigung der Übertragung

6.2.2 Funktionsweise

Die grundlegende Funktionsweise besteht darin, dass ein Client (meistens ein Webbrowser) eine Anfrage (**Request**) an einen Server stellt. Dafür muss der Client eine TCP-Verbindung aufbauen. Der Server antwortet darauf mit einem **Response**. Diese Antwort enthält bei http/1.1 immer einen Status-Code, der zum Beispiel besagt, ob die Transaktion erfolgreich abgeschlossen werden konnte. Zusätzlich werden Informationen über den Server übertragen und gegebenenfalls die angeforderten Daten. Danach beendet der Server die Verbindung.

Wenn man z.B. auf einer Website den Link http://www.beispiel.de/hompage.html anklickt, so wird an den **Computer** mit dem Namen **www.beispiel.de** die Anfrage gerichtet, die **Datei homepage.html** zurückzusenden. Der **Hostname** www.beispiel.de wird dabei zuerst über das DNS-Protokoll in eine **IP-Adresse** umgewandelt. Zur Übertragung wird über das TCP-Protokoll auf (meist) Port 80 eine **http-GET** Anforderung gesendet.

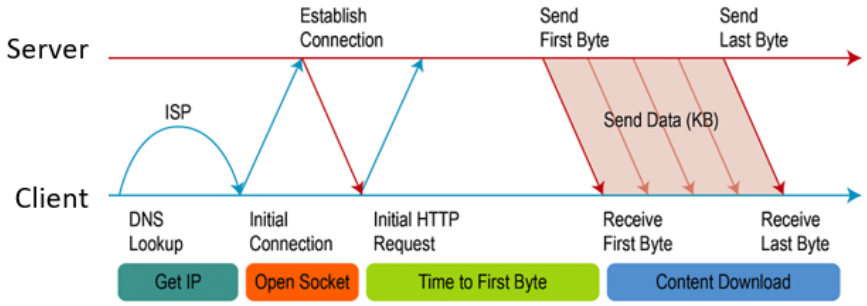


Abbildung 33 http Request

Das http-Protokoll ist **zustandslos**. Der http-Server kann keinen Zusammenhang zwischen einzelnen Anfragen eines Browsers herstellen. Jede Anfrage ist somit eigenständig und unabhängig von vorangegangenen Anfragen.

6.2.3 http-Request

Der folgende Request zeigt eine typische Anfrage eines Browsers an einen Webserver.

```
GET /homepage.html HTTP/1.1
Host: www.beispiel.de
Accept: text/html, image/jpeg, image/gif, */*
Accept-Charset: ISO-8859-1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:59.0) Gecko/20100101 Firefox/59.0
```

Jeder Request beginnt mit einer **Request-Line**. Diese Zeile beinhaltet zum ersten die **Methode**, also das was der Client vom Server möchte. In diesem Beispiel wird die **GET Methode**, die am häufigsten eingesetzte Methode, verwendet. Sie wird eingesetzt um Daten von einem Server anzufordern.

Der zweite Teil der Request-Line ist die **Request-URL** (/homepage.html). Hier gibt der Client an, welche Website oder Datei er aufrufen will. Der Host www.beispiel.de muss in den nachfolgenden Header-Feldern definiert werden.

Der dritte Teil der Request-Line gibt die verwendetet **HTTP-Version** an (http/1.1). Nach der Request-Line folgen die **Request Header**. Hier kann der Client zusätzliche Informationen über den Request und den Client selbst an den Server senden.

Die wichtigsten **Request-Methoden**:

GET:

Die Methode GET wird von einem Client eingesetzt, um die durch eine URL angegebenen Informationen (HTML-Dokumente, Bilder, Videos,...) anzufordern.

HEAD:

Die Methode HEAD hat die gleiche Aufgabe wie die Methode GET. Allerdings sendet der Server hier keine Daten zurücksenden, sondern nur den Antwort-Header. Dadurch kann ein Client etwa die Größe einer Datei oder die Verfügbarkeit einer Ressource abfragen.

POST:

ähnelt der GET-Methode, nur dass ein zusätzlicher Datenblock übermittelt wird. Dieser besteht üblicherweise aus Name/Wert-Paaren. Grundsätzlich können Daten auch mittels GET übertragen werden (als Parameter im URL), aber die zulässige Datenmenge ist bei POST deutlich größer.

PUT:

Die Methode PUT verlangt die Speicherung eines Dokuments unter der URL. Dabei können bestehende Dateien ersetzt werden, oder neue Dateien hinzugeschrieben werden, die unter einer neuen, bisher nicht existenten URL angelegt werden.

6.2.4 http-Response

Eine typische Antwort auf ein Request könnte folgendermaßen aussehen:

```
HTTP/1.1 200 OK
Date: Sat, 17-March-01 11:45:13 GMT
Server: Apache/1.3
Content-type: text/html
Content-length: 91

<html>
   <title>Hello</title>
   <body>
      Welcome to my world.
   </body>
</html>
```

Die erste Zeile ist hier die **Status-Line**. Sie enthält zum einen die **http-Version** und den **Status Code**. Dieser gibt das Ergebnis der Transaktion wieder und setzt sich zusammen aus einer dreistelligen Zahl und einer kurzen Beschreibung.

Die **http-Status-Codes** unterteilt man in die Bereiche:

- Codebereich 1xx: Allgemeine Informationen
- Codebereich 2xx: Anfrage des Clients verstanden und erfüllt
- Codebereich 3xx: Anfrage des Clients verstanden, jedoch nicht erfüllbar
- Codebereich 4xx: Anfrage des Clients unvollständig oder fehlerhaft
- Codebereich 5xx: Fehler im Server

Nach der Status-Line folgen die **Response-Header-Felder**. Diese enthalten Informationen über den Response, die nicht in der ersten Zeile untergebracht werden können. Dies können Informationen über den Server und die angeforderten Daten sein. Pro Zeile steht immer nur ein Header. Als Ende der Header-Felder steht genau eine Leerzeile.

Nach der Leerzeile, die dem letzten Header-Feld folgt, überträgt der Server die eigentlichen Informationen, beispielsweise den **HTML Code** oder die Bytes eines Bildes. Eine Response-Nachricht liefert im Normalfall Daten im Textformat. Sendet der Server Daten in einem anderen Format, wird das Format im Header angegeben.

6.2.5 http-Diagnose

Inzwischen besitzen gängige Webbrowser (*Internet Explorer*, *Chrome*, *Firefox*) die Möglichkeit Informationen über http und die übertragenen Inhalte anzuzeigen.

Anzeige im Webbrowser *Firefox*:

- Menü: ***Webentwickler / Netzwerkanalyse*** – öffnet Diagnosefenster
- Auswahl: ***Alles*** / ***Kopfzeilen*** *(unformatiert)*

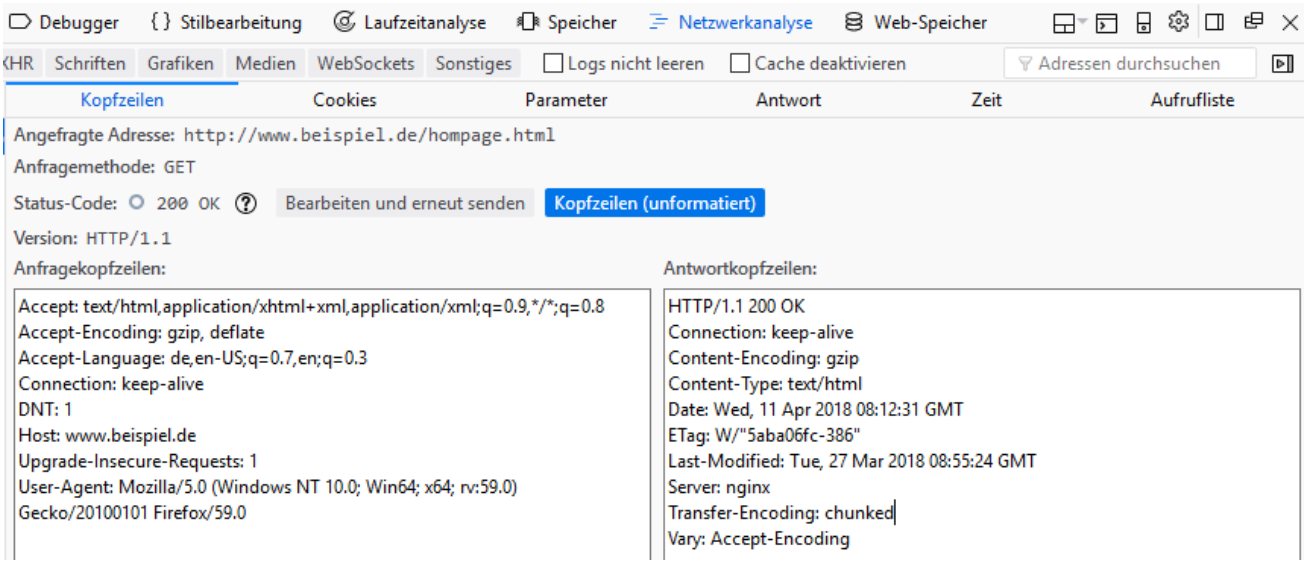


Abbildung 34 Anzeige der http-header in Firefox

6.2.6 Begriffe im Zusammenhang mit dem HTTP

Uniform Ressource Locator (**URL**): Ein URL gibt Übertragungsprotokoll, Server und Name für eine Datenquelle an.

Cascading Style Sheet (**CSS**): CSS ist eine Sprache für Stilvorlagen von strukturierten Dokumenten und wird bei der Erstellung von Benutzeroberflächen eingesetzt. Grundidee beim Entwurf von CSS war es, im Dokument bzw. Programm (z.B. HTML-Code) nur die strukturellen und inhaltlichen Teile zu beschreiben und mit CSS die visuelle Darstellung.

Common Gateway Interface (**CGI**): Unterstützt ein Server CGI-Scripts, so kann durch eine Client-Anfrage ein Programm (Script) am Server gestartet werden und der Output des Programms zum Client (Browser) übertragen werden. Typische Anwendung: Suchmaschine. Als Script-Sprache sind viele Sprachen verwendbar. Sehr verbreitet sind Pearl, PHP und Visual-Basic.

Cookie: Information, die der Server bei jedem Kontakt zum Client schickt und die vom Browser auf der Festplatte gespeichert wird. Dadurch kann der Server bei späteren Kontakten die "individuellen Bedürfnisse" des Client "befriedigen" (Haupanwendung: -> gezielte Werbung). Die Browser ermöglichen das Ablehnen von Cookies.

Weitere Infos zu http:
https://wiki.selfhtml.org/wiki/HTTP

6.3 URI: Unified Resource Identifier

Der **Unified Resource Identifier** oder „einheitlicher Bezeichner für Ressourcen“, ist eine eindeutige Kennzeichnung oder Adresse zum Auffinden einer Ressource, beispielsweise einer Webseite oder eine E-Mail-Adresse.
URI ist der **Überbegriff** für unterschiedliche Ressourcen wie:

- **URL - Uniform Resource Locator**
- **URN - Uniform Resource Names**

URLs und URNs sind URIs aber nicht umgekehrt.

Grundsätzlicher Aufbau eines URI: ***Schema* : *schemaspezifischer Teil***

Das Schema bestimmt den Aufbau des schemaspezifischen Teils. Der schemaspezifische Teil enthält die Angabe des Standorts oder einfach die Bezeichnung einer Ressource.
Liste aller festgelegten URIs:

http://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml

6.3.1 URL - Uniform Resource Locator

Zweck eines **URL** ist es, eine **Ressource zu lokalisieren**. Über die konkreten Inhalte und Formate werden keine Angaben gemacht. Verallgemeinernd wird eine URL meist als **Internetadresse** bezeichnet. Es können aber auch lokal vorliegende Dateien adressiert werden. Statt des Hostnamens ist auch eine IP-Nummer möglich.

Beispiel für **Schema http:**

```
http://max:muster@www.example.com:8080/index.html?p1=A&p2=B#ressource
\___/   \__/ \_____/ \_______________/ \__/ \_________/ \________/ \_______/
Schema  Benutzer  Kennwort       Hostname       Port      Pfad          Query      Fragment
```

Beispiel für **Schema mailto:**

```
mailto:max@example.org
\____/ \_______________/
Schema       e-Mail-Adresse
```

Beispiel für **Schema file:**

```
file:///verzeichnis/unterverzeichnis/datei
\___/ \_____________________________________/
Schema              lokale Datei
```

6.3.2 URN - Uniform Resource Name

Die URN bezeichnen ein Objekt mit einem eindeutigen Namen. Der Name enthält weder Informationen über Standort noch Inhalt, sondern eine **unverwechselbare Bezeichnung**.

Klassische URN-Objekte sind Produkte, die mit dem EAN-Code (European Article Number) ausgezeichnet sind, z.B. Bücher die den ISBN-Code tragen.

Beispiel eines URN:

```
URN:ISBN:3-8334-1681-5
```

6.4 HTML - Hyper Text Markup Language

Hypertext Markup Language (**HTML**), ist eine textbasierte Sprache zur Strukturierung digitaler Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten. HTML-Dokumente sind die Grundlage des World Wide Web und werden von Webbrowsern dargestellt.

HTML dient als dazu, einen Text semantisch zu strukturieren, nicht aber zu formatieren. Die visuelle Darstellung ist nicht Teil der HTML-Spezifikationen und wird durch den Webbrowser und Gestaltungsvorlagen wie ***Cascading Style Sheets*** (**CSS**) bestimmt. Mit CSS werden Gestaltungsanweisungen erstellt.

HTML wird **vom *World Wide Web Consortium*** (**W3C**) und der ***Web Hypertext Application Technology Working Group*** (**WHATWG**) weiterentwickelt. Die aktuelle Version ist seit dem 14. Dezember 2017
HTML 5.2.

6.4.1 HTML für Inhalt und Struktur

Im HTML-Quellcode werden mit Hilfe von HTML-Elementen der **Inhalt und die Struktur** festgelegt. Die Struktur ist die Logik des Texts:

- was ist eine Überschrift
- was ist der eigentliche Text
- was ist eine Aufzählung
- …

Der Aufbau jeder HTML Datei besteht aus **Informationen über den Inhalt** im head-Element und dem **eigentlichen Inhalt** im *body*-Element.

```
<!DOCTYPE html>
<html>
  <head>
    information about the page
  </head>

  <body>
    page contents
  </body>
</html>
```

HTML-Tag:

In Auszeichnungssprachen (engl. markup languages) wie beispielsweise XML und HTML, bezeichnen Tags die **Marken für Elemente**.

- `<Elementname>` – ein Starttag für den Beginn
- `</Elementname>` – ein Endtag für das Ende einer Auszeichnung

HTML-Element:

Beginnt mit einem **Starttag** und endet mit einem **Endtag**.

<Elementname>...</Elementname>

Beispiele:

Paragraph-Element repräsentiert einen Absatz: ** *Text* **

Überschrift-Element h1 .. h6 für sechs Hierarchieebenen: **<h1>*Überschrift*</h1>**

Zeilenumbruch-Element – besitzt keinen Inhalt und kein Endtag: **
**

Grafik-Element – besitzt keinen Inhalt und kein Endtag: **<img>**

HTML-Attribut:

Mit Attributen werden Elementen **Eigenschaften** zugewiesen. Attributwerte werden üblicherweise unter Hochkomma gesetzt.

<Elementname Attributname="Attributwert">

Beispiel:

Grafik-Element mit URL und Überschrift des Bilds: ****

HTML5 Elementdokumentation:

https://developer.mozilla.org/de/docs/Web/HTML/HTML5/HTML5_element_list

6.4.2 Design für Websites

Nach dem Inhalt und der Struktur einer Seite bekannt sind, wird das Design der Seite festgelegt. Das Design wird über mit **Cascading Style Sheets** (CSS), eine Sprache für Stilvorlagen von strukturierten Dokumenten, festgelegt.

In **CSS** werden Farben, Abstände, Rahmen und sonstige **Bestandteile des Erscheinungsbilds** definiert. Die Stärke von CSS ist, dass mit wenigen Befehlen, die alle untereinander kombiniert werden können, viele Design-Varianten möglich sind.

6.4.3 Beispiel einer HTLM- und CSS-Datei

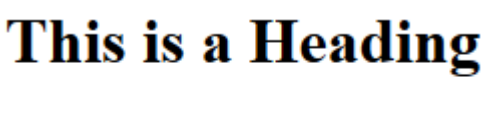
HTML-Datei:

```
<!DOCTYPE html>
<html>
    <head>
       <link rel="stylesheet" href="sheet3.css">
       <title>Page Title</title>
    </head>

    <body>
       <h1>This is a Heading</h1>
       <p>This is a paragraph. %03d</p>
    </body>
</html>
```

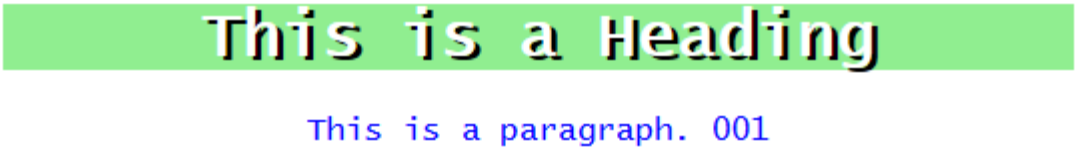
CSS-Datei (sheet3.css):

```
h1 {
    font-family: "lucida console", sans-serif;
    text-shadow: 3px 2px black;
    text-align: center;
    background-color: lightgreen;
    color: white;
}
p {
    font-family: "lucida console", sans-serif;
    text-align: center;
    color: blue;
}
```



This is a paragraph. 013

Abbildung 35 Anzeige ohne CSS



This is a paragraph. 001

Abbildung 36 Anzeige mit CSS (sheet3.css)

HTML-Tutorials:

https://www.w3schools.com/html/

HTML-Validator:

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default