

# Grundlagen

## 1.1 Minimalismus und Autonomie

- ↳ Netzwerk soll in der Lage sein selbstständig zu arbeiten, ohne dass interne Änderungen notwendig sind, um mit anderen Netzwerken zusammen zu arbeiten

### Best-Effort-Dienst

- ↳ End-zu-End Verbindung zur Kommunikation.
- ↳ Verlorene Pakete sollen Neuübertragen werden.

### Zustandslose Router

- ↳ Keine Informationen über das Netzwerk und die laufenden Verbindungen

### Dezentrale Kontrolle

- ↳ keine globale Kontrolle der Netzwerke

## 1.2 Das Internet Protokollstack

<u>Schicht/Ebene</u>	<u>Protokoll</u>	<u>Aufgabe</u>
Anwendungsschicht	HTTP, FTP, SMTP	Unterstützung der Netzworkeanwendung
Transportschicht	TCP, UDP	Datentransfer von Host-zu-Host
Internetschicht	IP, Routing Protokolle	Routing von Datagrammen von der Quelle zum Ziel
Netzzugangsschicht Netzwerkschicht	PPP, Ethernet	Datentransfer zwischen benachbarten Netzwerkelementen

1.3 - Sicherheitschicht des OSI-Modells, bietet IP seine Dienste dem TCP/IP an.

- IP: Routing für TCP/UDP
- ~~TCP: paketorientiert~~
- TCP/UDP siehe 1.4



1.4 TCP	UDP
<u>Verbindungsorientiert</u>	<u>Verbindungsloses Transportprotokoll</u>
<u>Flusskontrolle / Überlastungskontrolle (großer Header)</u>	<u>Unzuverlässiger Datentransfer</u>
<u>Bietet keine Zeit- und Bandbreitengarantie</u>	<u>Schneller als TCP</u>
<u>Verwendung für FTP, HTTP, SMTP</u>	<u>Verwendet beim Streaming, Internet-Telefonie</u>

1.5 Mit einer IP wird ein Host eindeutig in einem Netzwerk identifiziert.

Da Zahlen so schwer zu merken sind, benutzt man Domänen.

Durch den DNS wird die Domänenadresse in eine IP umgewandelt

IPv4 (4 Byte) <sup>48 Bit (2<sup>48</sup>)</sup> ⇒ Header mit TTL, Source, Destination, Länge (20 Byte)

IPv6 (16 Byte) <sup>128 Bit (2<sup>128</sup>)</sup> ⇒ <sup>8 Blöcke ⇒ Hex</sup>

Durch die Subnetmask wird aus der IP der Host und Netzwerk-teil Anteil heraus gelesen.

1.6 Identifikator für eine abstrakte oder physische Ressource (Obermenge von URLs)

- Reihenfolge

- Schema, Anbieter, Pfad, ASfrage, Fragment

URL



Grundlagen	Prinzipien
------------	------------

[illegible]

- 1.7
  - ↳ Zustandslos, auf Basis von TCP, 'CRUD, REST'
  - ↳ Request/Response Prinzip, Übertragung im Klartext
  - ↳ HTTP Request
    - ↳ get/Head/Post/PUT/Delete

Informationen: URI    ohne  
Backspace    center  
URI / Header    update    location

### ↳ HTTP Response

Bestandteile

- ↳ Status: 200 Document follows  
403 Forbidden  
404 Not Found
- ↳ Metainfos: Datum, Server-Software, Dokumentlänge
- ↳ eigentliches Dokument

1.8 Wenn es ist Zustandslos

- ↳ Konsequenzen: Keine Flusskontrolle Server weiß nicht  $\rightarrow$  die Response ankommt.

1.9 #Get/Head/Post/Put/Delete

1.10 Get: Informationen stehen in der URL

POST  $\rightarrow$  Header: Infos im Header, & Information kann beliebig lang sein, muss im Header & angegeben werden

GFT: Lesen der durch URL gegebenen Ressource

POST: Erzeugen einer Ressource unter der gegebenen ~~Ressource~~ URI

POST URL

## HTTP-Header

$$key_0 = \text{value}_0 \quad \& \quad key_n = \text{value}_n, \dots$$

↑  
Trainer