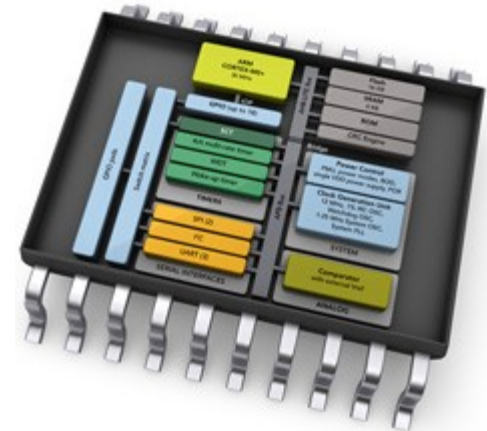


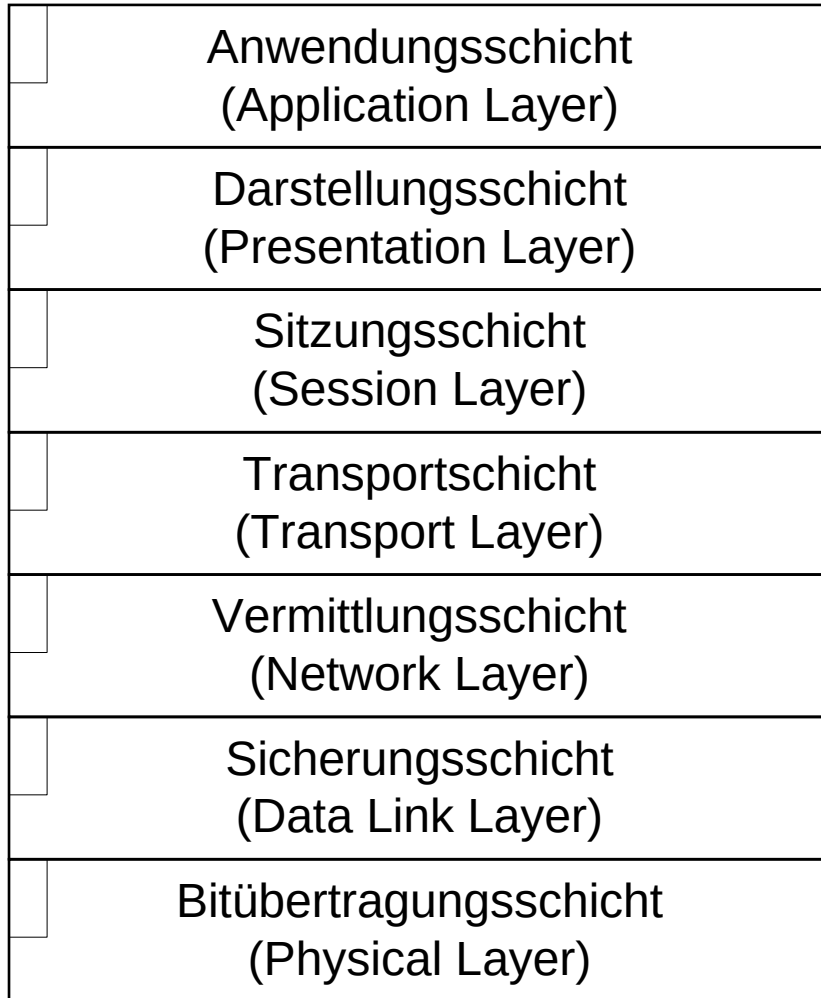
Kommunikationssysteme

(Modulcode 941306)

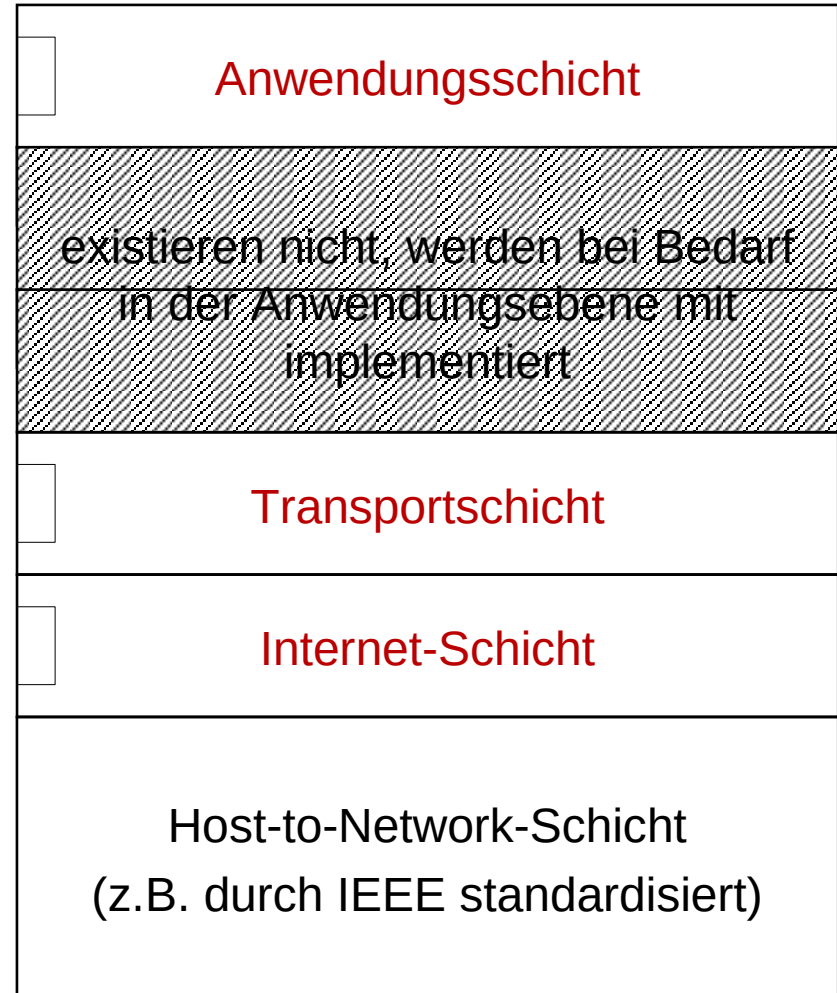
Prof. Dr. Andreas Terstegge



Das Internet-Referenzmodell

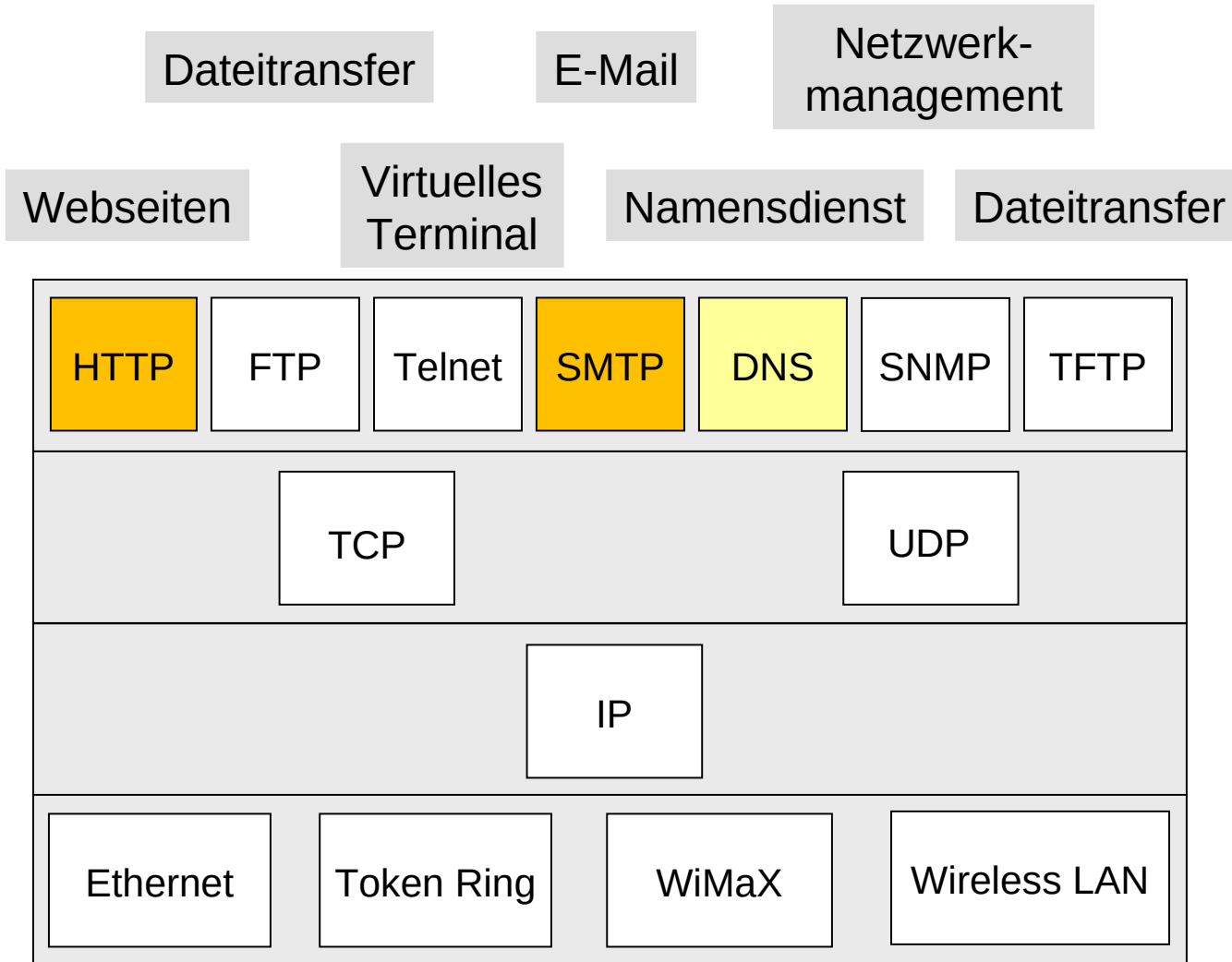


ISO/OSI



TCP/IP

Anwendungsprotokolle im TCP/IP-Referenzmodell



Anwendungsorientierte
Protokolle

Portnummern

IP-Adressen

MAC-Adressen

Netzwerkprogrammierung:

- TCP/UDP und IP sind im Betriebssystem integriert
- Wird eine eigene Anwendung entwickelt, hat man kein vorgefertigtes “Anwendungsprotokoll”, sondern muss direkt auf TCP/IP aufbauen
- Port-Nummern identifizieren die Anwendung, IP-Adressen den Zielrechner
- Zugang zu TCP/UDP/IP über **Sockets**

Hinweis: Selbstverständlich gibt es Bibliotheken, die auf Basis von TCP/IP höherwertige Kommunikationsprimitiven bieten (Middleware)

Sockets:

- Abstraktion, die vom Betriebssystem in den Programmiersprachen bereitgestellt wird
- Teilt man einer Instanz dieses Typs Adressinformationen des Kommunikationspartners mit, wird ein *Kommunikationskanal* zu diesem Partner aufgebaut (TCP), bzw. vorbereitet (UDP)
- Bei **TCP** kann man einfach durch schreiben auf den Socket bzw. auslesen des Sockets Daten übertragen; In Java wird dies durch **Streams** realisiert
- Bei **UDP** muss man quasi eine Art **Briefkasten** benutzen, über den die Nachricht (mit ausgewiesenem Absender und Empfänger) übertragen wird

Aber: der Programmierer ist verantwortlich dafür, was übertragen wird!

Arbeiten mit TCP/IP: Client (schematisch)

Angabe der Adresse des Servers in zwei Teilen:
Adresse des Zielrechners (IP-Adresse/Name) und der
auf diesem Rechner laufenden Anwendung, die die
Daten bekommen soll (Port). Diese Adressinformationen
werden mit einem sogenannten Socket verknüpft, der
eine Variable zur Kommunikation per TCP/IP darstellt.

```
import java.io.*;  
import java.net.*;  
class TCPClient {
```

```
    public static void main(String argv[]) throws Exception  
    {
```

```
        String sentence;  
        String modifiedSentence;
```

```
        BufferedReader inFromUser = new BufferedReader  
            (new InputStreamReader(System.in));
```

Erstelle Client-
Socket, baue
Verbindung auf

```
        Socket clientSocket = new Socket("zielrechner", 6789);
```

Erstelle
Datenstrom
für den Socket

```
        DataOutputStream outToServer = new DataOutputStream  
            (clientSocket.getOutputStream());
```

Arbeiten mit TCP/IP: Client (schematisch)

Erstelle
Datenstrom
aus dem
Socket

```
BufferedReader inFromServer = new BufferedReader  
    new InputStreamReader(clientSocket.getInputStream());
```

Sende an
den Server

```
sentence = inFromUser.readLine();  
outToServer.writeBytes(sentence + '\n');
```

SEND

Empfange
vom Server

```
modifiedSentence = inFromServer.readLine();  
System.out.println("FROM SERVER: " + modifiedSentence);  
clientSocket.close();
```

RECEIVE

```
}  
}
```

Arbeiten mit TCP/IP: Server (schematisch)

```
import java.io.*;  
import java.net.*;
```

```
class TCPServer {
```

```
    public static void main(String argv[]) throws Exception  
    {
```

```
        String clientSentence;  
        String capitalizedSentence;
```

Erstelle
Socket]

```
        ServerSocket welcomeSocket = new ServerSocket(6789);
```

Warte auf
eingehende
Verbindungs-
wünsche]

```
        while(true) {
```

```
            Socket connectionSocket = welcomeSocket.accept();
```

```
            BufferedReader inFromClient =  
                new BufferedReader(new
```

Verknüpfe Buffer
mit dem Socket]

```
                InputStreamReader(connectionSocket.getInputStream()));
```

Wie beim Client: erstelle einen Socket als Variable zur Kommunikation. Aber: lege direkt fest, über welchen Port diese Anwendung Daten empfangen soll.
Warte dann so lange, bis ein Client an diesen Port Daten schickt.

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

```
DataOutputStream outToClient =  
    new DataOutputStream(connectionSocket.getOutputStream());
```

```
clientSentence = inFromClient.readLine();
```

RECEIVE

```
capitalizedSentence = clientSentence.toUpperCase() + '\n';
```

```
outToClient.writeBytes(capitalizedSentence);
```

Gehe zurück, warte auf nächste Anfrage

Der eben implementierte Server hat ein Problem!

Welches?

FH Aachen
Fachbereich 9 Medizintechnik und Technomathematik
Prof. Dr.-Ing. Andreas Terstegge
Straße Nr.
PLZ Ort
T +49. 241. 6009 53813
F +49. 241. 6009 53119
Terstegge@fh-aachen.de
www.fh-aachen.de