

# Kommunikationssysteme WS24/25

## Übungsblatt 1

Maximilian Amthor

# Aufgabe 1.1

# Aufgabe 1.1

- a) Erklären Sie die Unterschiede zwischen „Internet“ und „WWW“. Ordnen Sie jedem der beiden Begriffe je 2 genutzte Protokolle zu.
  
- b) Erklären Sie den Begriff der Interoperabilität, sowie dessen Bedeutung für die Funktionsfähigkeit des Internets.

# Lösung zu 1.1 a)

## □ Internet

- ❖ Infrastruktur zur Übertragung von Daten zwischen Computern und Netzwerken
- ❖ generischer Kommunikationsservice
- ❖ Protokolle TCP, IP, UDP (TCP/IP-Stack)

## □ WWW

- ❖ Nutzt den Kommunikationsservice Internet
- ❖ reine Anwendung
- ❖ Implementierung durch Web-Server und Web-Browser
- ❖ Protokolle HTTP, HTTPS, DNS

# Aufgabe 1.1

- a) Erklären Sie die Unterschiede zwischen „Internet“ und „WWW“. Ordnen Sie jedem der beiden Begriffe je 2 genutzte Protokolle zu.
- b) Erklären Sie den Begriff der Interoperabilität, sowie dessen Bedeutung für die Funktionsfähigkeit des Internets.

# Lösung zu 1.1 b)

- ❑ **Interoperabilität**

= Fähigkeit verschiedener Systeme miteinander zu kommunizieren

- ❑ **Netzwerk-Protokoll** enthält Regeln, die alle Aspekte der Kommunikation festlegen

- ❖ z.B. elektrische Spannung, Verbindungsauf-/abbau, Fehlerbehandlung,...

- ❑ ermöglicht Kommunikation zwischen verschiedenen Systemen und Plattformen → **Internet**

## Aufgabe 1.2

# Aufgabe 1.2

- a) Unterscheiden Sie die beiden grundlegenden Kommunikationsparadigmen. Gehen Sie dabei insbesondere auf die Rollenverteilung ein. Welchem dieser beiden Paradigmen würden Sie Google Fast Share zuordnen?
- b) Welche Schritte durchläuft ein Client, wenn er einen Server unter einer bestimmten Domain anfragt? Welche Schritte werden beim Server durchlaufen?
- c) Welches Betriebssystemfeature nutzt ein Server, um mehrere Verbindungen parallel abzuarbeiten?
- d) Überlegen Sie sich, welche Probleme entstehen können, wenn es zyklische Abhängigkeiten zwischen Servern gibt?



# Lösung zu 1.2 a)

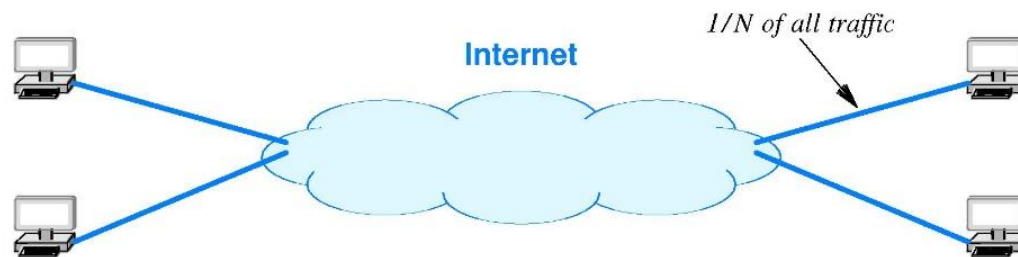
## □ Client-Server Modell

Server Anwendung	Client Anwendung
Startet zuerst	Start als zweites
Kennt die potentiellen Clients nicht	Muss den Server kennen
Wartet passiv auf Verbindungen von Clients	Baut aktiv die Verbindung zum Server auf
Empfängt Daten vom und sendet Daten zum Client	Sendet Daten zum und empfängt Daten vom Server
Bearbeitet beliebig viele Clients nacheinander und teils parallel	Beendet sich möglicherweise nach erfolgreicher Kommunikation

# Lösung zu 1.2 a)

## □ Peer-to-Peer Modell

- ❖ keine Unterteilung in Client und Server
- ❖ jeder Peer kann Dienste anfordern und bereitstellen
- ❖ Teilnehmer speichern und verwalten eigene Daten
- ❖ nicht zwingend 1:1-Verbindung



## Lösung 1.2 a)

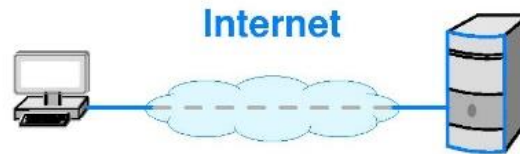
- ❑ **Google Fast Share** ermöglicht unkomplizierten, lokalen Datenaustausch
- ❑ Geräte kommunizieren direkt miteinander ohne vermittelnden Server → **Peer-to-Peer**

# Aufgabe 1.2

- a) Unterscheiden Sie die beiden grundlegenden Kommunikationsparadigmen. Gehen Sie dabei insbesondere auf die Rollenverteilung ein. Welchem dieser beiden Paradigmen würden Sie Google Fast Share zuordnen?
- b) Welche Schritte durchläuft ein Client, wenn er einen Server unter einer bestimmten Domain anfragt? Welche Schritte werden beim Server durchlaufen?
- c) Welches Betriebssystemfeature nutzt ein Server, um mehrere Verbindungen parallel abzuarbeiten?
- d) Überlegen Sie sich, welche Probleme entstehen können, wenn es zyklische Abhängigkeiten zwischen Servern gibt?

# Lösung zu 1.2 b)

- Start after server is already running
- Obtain server name from user
- Use DNS to translate name to IP address
- Specify the port that the service uses, N
- Contact server and interact



- Start before any of the clients
- Register port N with the local system
- Wait for contact from a client
- Interact with client until client finishes
- Wait for contact from the next client

# Aufgabe 1.2

- a) Unterscheiden Sie die beiden grundlegenden Kommunikationsparadigmen. Gehen Sie dabei insbesondere auf die Rollenverteilung ein. Welchem dieser beiden Paradigmen würden Sie Google Fast Share zuordnen?
- b) Welche Schritte durchläuft ein Client, wenn er einen Server unter einer bestimmten Domain anfragt? Welche Schritte werden beim Server durchlaufen?
- c) Welches Betriebssystemfeature nutzt ein Server, um mehrere Verbindungen parallel abzuarbeiten?
- d) Überlegen Sie sich, welche Probleme entstehen können, wenn es zyklische Abhängigkeiten zwischen Servern gibt?

# Lösung zu 1.2 c)

- ❑ Server nutzen **Threads** um mehrere Clients gleichzeitig zu bearbeiten
- ❑ Ein **Haupt-Thread** zur Annahme neuer Clients
  - ❖ blockiert sobald Thread-Pool leer
- ❑ Ein **Handler-Thread** pro akzeptierter Verbindung
  - ❖ werden Thread-Pool entnommen
  - ❖ bearbeitet Daten/ Requests des Client
  - ❖ kann nach Verbindungsabbau neuen Client bedienen

# Exkurs Implementierung

## ❑ Server

```
while (true) {  
    var socket = serverSocket.accept();  
    System.out.println(socket.getLocalPort());  
    pool.execute(new Handler(socket));  
}
```

## ❑ Client

```
var host = "localhost";  
var port = 50000;  
  
try (var socket = new Socket(host, port);
```



# Aufgabe 1.2

- a) Unterscheiden Sie die beiden grundlegenden Kommunikationsparadigmen. Gehen Sie dabei insbesondere auf die Rollenverteilung ein. Welchem dieser beiden Paradigmen würden Sie Google Fast Share zuordnen?
- b) Welche Schritte durchläuft ein Client, wenn er einen Server unter einer bestimmten Domain anfragt? Welche Schritte werden beim Server durchlaufen?
- c) Welches Betriebssystemfeature nutzt ein Server, um mehrere Verbindungen parallel abzuarbeiten?
- d) Überlegen Sie sich, welche Probleme entstehen können, wenn es zyklische Abhängigkeiten zwischen Servern gibt?

# Lösung zu 1.2 d)

- ❑ Server für einen Service kann Client eines anderen sein
  - ❖ z.B. Web-Server wird Client einer Datenbank oder eine Sicherheitsdienstes
- ❑ unendliche Schleifen
  - ❖ Server A wird Client von Server B, B von C und C von A
  - ❖ Request-Kette läuft unendlich
- ❑ Single-Point of Failure

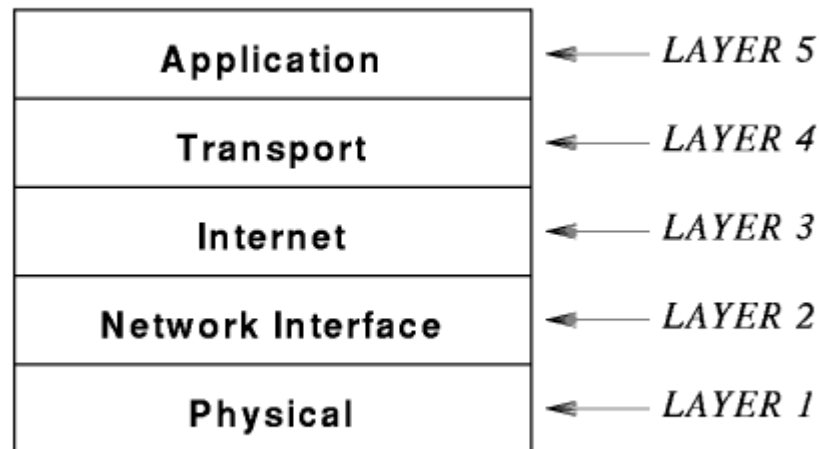
## Aufgabe 1.3

# Aufgabe 1.3

- a) Erläutern Sie das Internet Layering Model (TCP/IP Schichtenmodell)!  
Welche Schichten werden unterschieden und welche Aufgaben erfüllen sie?
  
- b) Welche Protokolle kennen Sie und in welche Schichten ordnen Sie diese ein?

# Lösung zu 1.3

## □ TCP/IP Schichtenmodell:



# Lösung zu 1.3

## □ TCP/IP Schichtenmodell:

### ❖ **Application:**

- Regelt die Kommunikation zwischen Applikationen
- Protokolle: HTTP, FTP, SMTP
- Entspricht ISO Schichten 5 - 7 (Session, Presentation und Application)

### ❖ **Transport:**

- Spezifiziert (zuverlässigen) Transfer von Daten
- Protokolle: UDP, TCP
- Entspricht ISO Schicht 4 (Transport)

# Lösung zu 1.3

## □ TCP/IP Schichtenmodell:

### ❖ Internet:

- Definiert das Format von Paketen und Mechanismen zum Weiterleiten von Paketen von einem Computer über mehrere Router zum Ziel
- Protokoll: IP
- Entspricht ISO Schicht 3 (Network)

### ❖ Network Interface:

- Definiert Formate zur Übertragung von Paketen in Hardwarerahmen (Frames)
- Protokolle: LLC (dort u.a. ARP), MAC (dort u.a. CSMA/CD)
- Entspricht ISO Schicht 2 (Data Link)

# Lösung zu 1.3

## □ TCP/IP Schichtenmodell:

### ❖ Physical:

- Definiert grundlegende Netzwerk Hardware
- Protokolle: RS 232, V.24, X.21
- Entspricht ISO Schicht 1 (Physical)



## Aufgabe 1.4

# Aufgabe 1.4

Zwei Netze bieten zuverlässigen verbindungsorientierten Dienst. Eines davon bietet einen zuverlässigen **Bytestrom** und das andere einen zuverlässigen **Nachrichtenstrom**. Sind diese identisch?

Falls ja, warum wird diese Unterscheidung gemacht? Falls nicht, geben Sie ein Beispiel an, um den Unterschied zu verdeutlichen.

# Lösung zu 1.4

## □ **Stream Transport**

- ❖ verbindungsorientierte Übertragung von Byte-Strömen
- ❖ kontinuierlicher Datenstrom ohne klare Abgrenzung
- ❖ bidirektional (Daten fließen in beide Richtungen)
- ❖ nutzt TCP

## □ **Message Transport**

- ❖ verbindungslose Übertragung von Nachrichten
- ❖ max. 64KB Nachrichten
- ❖ keine Kompensation von Verlust, Vertauschung,...
- ❖ ermöglicht 1:N und N:M Kommunikation
- ❖ nutzt UDP

# Lösung zu 1.4

- somit sind die beiden Netze **nicht identisch**, bezogen auf die Transportschicht (Layer 4)
  
- **Beispiel Videoübertragung:**
  - ❖ Video ist Abfolge von Frames, die nacheinander abgespielt werden
  - ❖ zuverlässiger Bytestrom ermöglicht nahtlose Übertragung ohne Unterbrechungen
  - ❖ zuverlässiger Nachrichtentransport unterteilt Video in Pakete, die wieder zusammengesetzt werden müssen

# Lösung 1.4

## □ **Beispiel Chat-Dienst:**

- ❖ Nachrichten sind diskrete Einheiten (Dateien, Textnachricht, Bild,...)
- ❖ zuverlässiger Nachrichtenstrom ermöglicht unabhängige Übertragung und einfache Verarbeitung
- ❖ zuverlässiger Bytestrom kann zu erhöhter Latenz durch Puffer führen

## □ **Anmerkung: Bezogen auf die Internetschicht (Layer 3) können beide Netzwerke auch als **identisch** angesehen werden**

- ❖ jeweils Kapselung in IP-Datagramme

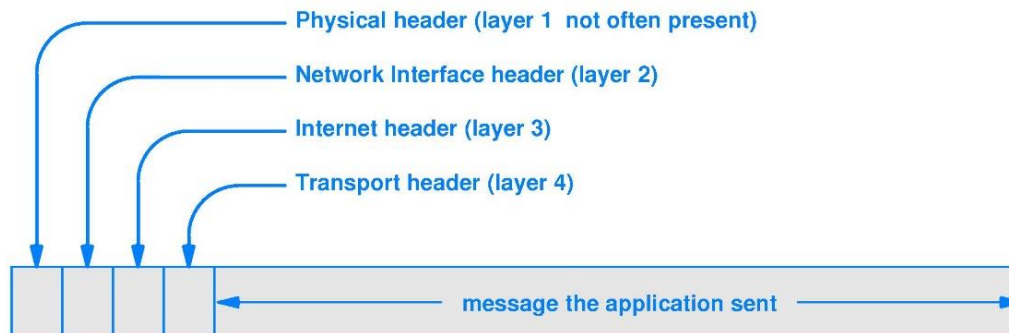
## Aufgabe 1.5

# Aufgabe 1.5

- ❑ Ein System habe eine Protokollhierarchie mit 5 Schichten
- ❑ Anwendungen erzeugen Nutzdaten mit einer Länge von 100 Bytes
- ❑ Auf jeder Schicht 10 Byte Header
- ❑ Frage: Wie groß ist der Protokolloverhead in Prozent?

# Packet Assemblierung

- ❑ Jedes Protokoll jeder Schicht ergänzt beim Senden **Metadaten** zu den zu transportierenden Daten
  - ❖ Prüfsummen
  - ❖ Adressen (Ports, IP-Adressen, MAC-Adressen)
  - ❖ Kontrollinformationen (Paketnummern, Segmentierung)
- ❑ Header mit diesen Daten wird in jeder Schicht den Daten **voran gestellt**
- ❑ Beim Empfänger extrahiert jedes Protokoll seinen entsprechenden Header und wertet Daten aus





# Lösung zu 1.5

## □ Allgemein:

- ❖  $n$  Schichten
- ❖  $m$  Bytes Nutzdaten
- ❖  $h$  Bytes Header pro Schicht:
- ❖ Overhead in Bytes:  $h * n$
- ❖ Gesamtpaketlänge:  $h * n + m$
- $((h * n) / (h * n + m)) * 100 = \text{Protokoll-Overhead in \%}$

## □ Lösung:

- ❖  $n = 5$  Schichten;  $m = 100$  Bytes Nutzdaten;
- ❖  $h = 10$  Bytes Header pro Schicht
- $5 * 10 / (5 * 10 + 100) = 50 / 150 = 1/3$  (**33%**)

## Aufgabe 1.6

# Aufgabe 1.6

Ordnen Sie folgende Use-Cases den verantwortlichen Komponenten einer Android Applikation zu:

- a) Es soll immer eine Benachrichtigung erscheinen, wenn ein neues Fußballergebnis erscheint.
- b) Ich möchte ein Bild von meiner Kamera mit einer Messenger-App teilen.
- c) Nutzung des Exposure Notifikation Frameworks zum Corona Kontakt-Tracing.
- d) Eine Video-App soll das Video stoppen, wenn ein Anruf eingeht

# Android-Komponenten

## □ **Activity**

- ❖ GUI-Elemente für Nutzerinteraktion (z.B. Kontaktliste)

## □ **Service**

- ❖ Hintergrundprozesse (z.B. Abspielen von Musik)

## □ **Content Provider**

- ❖ Austausch von Daten zwischen Anwendungen (z.B. Kalender)

## □ **Broadcast Receiver**

- ❖ systemweite Ereignisse (z.B. neue Zeitzone)

# Lösung zu 1.6

## a) **Service, Activity**

- ❖ Überprüfung aktueller Fußballergebnisse im Hintergrund
- ❖ Anzeigen und Interaktion mit der Meldung

## b) **Content Provider, Activity**

- ❖ Bild wird gespeichert und Messenger App zur Verfügung gestellt
- ❖ Anzeigen von Bild und Oberfläche zur Weiterleitung

## c) **Service**

- ❖ Überwachung Bluetooth Verbindung im Hintergrund

# Lösung zu 1.6)

## d) **Broadcast Receiver, Activity**

- ❖ Erfassen des eingehenden Anrufes durch Broadcast Receiver
- ❖ Activity des Videos wird gestoppt