

# Übungsblatt 3 (Musterlösung)

## 1. Terminologie

Bitte grenzen Sie die folgenden Begrifflichkeiten voneinander ab:

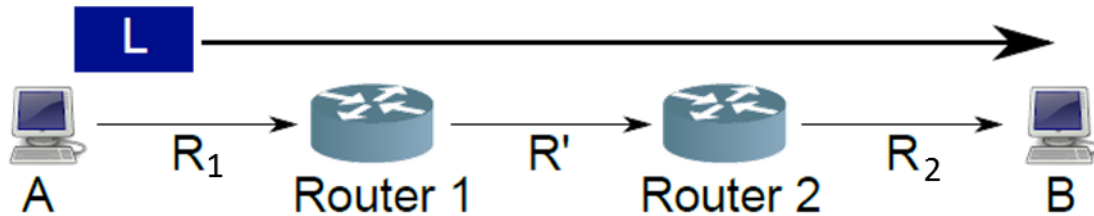
- unidirektional
- bidirektional
- simplex
- half-duplex
- full-duplex
- duplex
- unicast
- broadcast
- multicast
- geocast
- anycast

## Lösung

- Unidirektional ist der Betrieb in ausschließlich eine Richtung. Beispiele sind Radio, Fernsehen, Videotext, ...
- Bidirektional ist der Betrieb in beide Richtungen. Beispiele sind Telefon, Chat, E-Mail, ...
- Simplex ist der gerichtete Betrieb. Kein wesentlicher Unterschied zu unidirektional.
- Half-duplex ist der wechselseitig gerichtete Betrieb. Ein Beispiel ist Amateurfunk.
- Full-duplex ist der gleichzeitige Betrieb. Beispiele sind Onlinespiele, DSL, hitzige Gespräche, ...
- Duplex ist der Oberbegriff. Es wird auch keine Aussage darüber getroffen, ob Senden und Empfangen gleichzeitig möglich ist. Nach einer anderen Definition ist duplex gleichbedeutend mit full-duplex.
- Unicast bedeutet, dass genau ein Teilnehmer angesprochen wird. Z.B. Telefon, aber keine Telefonkonferenz.
- Broadcast bedeutet, dass all Teilnehmer adressiert werden. Hier ist z.B. Radio als Beispiel zu nennen.
- Multicast bedeutet, dass mehrere benannte Teilnehmer am Empfang teilnehmen. Beispielsweise sei hier Pay-TV genannt.
- Geocast bedeutet, dass der Rundruf örtlich begrenzt ist. Beispiele sind hier z.B. Vorlesung (ohne Aufzeichnung), Bierzeltmusik
- Anycast bedeutet, dass irgendwer gemeint ist. Hier kann man den Hilferuf als Beispiel nennen.

## 2. Paketvermittlung - Delays

Es sei das in der Abbildung gezeigte Netzwerk gegeben. Dabei schickt Alice A ein Paket an Bob B.



Das Netzwerk hat folgende Eigenschaften:

- Die Größe des Pakets ist 100 kbyte.
- $R_1$  und  $R_2 = 1\,000\text{ kbits/s}$ ,  $R' = 10\,000\text{ kbits/s}$ .
- Die Router verwenden Store-and-Forward.
- Processing Delay: Router 1: 0,22 ms, Router 2: 0,1 ms
- Queueing Delay im Router 1: 10 ms, Router 2: 20 ms
- Propagation Delay  $R_1 = 1\text{ ms}$ ,  $R' = 10\text{ ms}$ ,  $R_2 = 2\text{ ms}$

Wie lange dauert es, bis das Paket von Alice an Bob übertragen worden ist?

### Lösung

Betrachtung für 1 Paket mit 100 kbyte = 800 kbits:

Verzögerung	$R_1$	Router 1	$R'$	Router 2	$R_2$
Queueing		10 ms		20 ms	
Transmission	$\frac{800\text{ kbits}}{1\,000\text{ kbits/s}}$		$\frac{800\text{ kbits}}{10\,000\text{ kbits/s}}$		$\frac{800\text{ kbits}}{1\,000\text{ kbits/s}}$
Propagation	1 ms		10 ms		2 ms
Processing		0,22 ms		0,1 ms	

Gesamtverzögerung für das Paket:

Queueing: 30 ms

Transmission: 0,8 ms + 0,08 ms + 0,8 ms

Propagation: 13 ms

Processing: 0,32 ms

Gesamt: 45 ms

## 3. HTTP-Anwortzeiten

Es soll mit einem Webbrowser eine Homepage mit vier Bildern angezeigt werden. Während die HTML-Datei eine Größe von 15 kByte hat, sind zwei der Bilder jeweils 200 kByte groß und die anderen beiden jeweils 240 kByte. Gehen Sie davon aus, dass die Transmission Delays aller weiteren übertragenen Daten (wie GET-Requests, TCP-Handshake, alle Header Informationen) vernachlässigt werden können. Es soll also der Transmission Delay nur bei den Daten berücksichtigt werden. Die Verbindung zum Server hat eine Bandbreite von 2 MBit/s und eine zu erwartende RTT (Round-Trip-Time) von 200 ms. Beachten Sie für die

folgenden Aufgaben, dass persistentes HTTP ohne und mit Pipelining durchgeführt werden kann. Wie lange dauert es, bis die Homepage inklusive der Bilder angezeigt werden kann, wenn man

- nichtpersistentes http
- persistentes HTTP ohne Pipelining
- persistentes HTTP mit Pipelining verwendet?

## Lösung

### *HTTP Antwortzeiten – nichtpersistentes HTTP*

- Es sind die vier Bilder und die HTML-Seite an sich runterzuladen. Damit haben wir 5 Handshakes. Dazu kommen die fünf GET-Requests und deren Antworten. Beachten Sie, dass hier die RTT mit 200ms angegeben wurde. Nun haben wir eine Bandbreite von 2MBit/s. Damit haben wir alles zusammen, um die Dauer zu berechnen. Es sei nun

$$d_{\text{Handshake}} = 5 \cdot \text{RTT} = 5 \cdot 200\text{ms} = 1000\text{ms} = 1\text{s}$$

$$d_{\text{GET-Requests}} = 5 \cdot \text{RTT} = 5 \cdot 200\text{ms} = 1000\text{ms} = 1\text{s}$$

$$d_{\text{Übertragung}} = \frac{895\text{kByte}}{2\text{MBit/s}} = \frac{7,16\text{MBit}}{2\text{MBit/s}} = 3,58\text{s}$$

Damit gilt:

$$d_{\text{Gesamt}} = d_{\text{Handshake}} + d_{\text{GET-Requests}} + d_{\text{Übertragung}} = 5,58\text{s}$$

### *HTTP Antwortzeiten – persistent ohne Pipelining*

- Es sind die vier Bilder und die HTML-Seite an sich runterzuladen. Da wir nun persistentes HTTP verwenden, haben wir nur einen Handshake. Dazu kommen allerdings die fünf GET-Requests und deren Antworten, da wir kein Pipelining verwenden. Beachten Sie, dass hier die RTT mit 200ms angegeben wurde. Nun haben wir eine Bandbreite von 2MBit/s. Damit haben wir alles zusammen, um die Dauer zu berechnen. Es sei nun

$$d_{\text{Handshake}} = 1 \cdot \text{RTT} = 1 \cdot 200\text{ms}$$

$$d_{\text{GET-Requests}} = 5 \cdot \text{RTT} = 5 \cdot 200\text{ms} = 1000\text{ms} = 1\text{s}$$

$$d_{\text{Übertragung}} = \frac{895\text{kByte}}{2\text{MBit/s}} = \frac{7,16\text{MBit}}{2\text{MBit/s}} = 3,58\text{s}$$

Damit gilt:

$$d_{\text{Gesamt}} = d_{\text{Handshake}} + d_{\text{GET-Requests}} + d_{\text{Übertragung}} = 4,78\text{s}$$

### **HTTP Antwortzeiten – persistent mit Pipelining**

- Es sind die vier Bilder und die HTML-Seite an sich runterzuladen. Da wir nun persistentes HTTP verwenden, haben wir nur einen Handshake. Nun kommt allerdings nur ein GET-Request für die HTML-Seite und einer für alle Bilder (selbstverständlich inklusive Antworten), da wir Pipelining verwenden. Beachten Sie, dass hier die RTT mit 200ms angegeben wurde. Nun haben wir eine Bandbreite von 2MBit/s. Damit haben wir alles zusammen, um die Dauer zu berechnen. Es sei nun

$$d_{\text{Handshake}} = 1 \cdot \text{RTT} = 1 \cdot 200\text{ms}$$

$$d_{\text{GET-Requests}} = 2 \cdot \text{RTT} = 2 \cdot 200\text{ms} = 400\text{ms}$$

$$d_{\text{Übertragung}} = \frac{895\text{kByte}}{2\text{MBit/s}} = \frac{7,16\text{MBit}}{2\text{MBit/s}} = 3,58\text{s}$$

Damit gilt:

$$d_{\text{Gesamt}} = d_{\text{Handshake}} + d_{\text{GET-Requests}} + d_{\text{Übertragung}} = 4,18\text{s}$$

**Viel Erfolg!!!**